1996-09

Error probabilities of spread spectrum systems in [an] ultra wideband source (UWBS) interferenc

Yeu, Eng-Kiong.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

THESIS

ERROR PROBABILITIES OF SPREAD SPECTRUM SYSTEMS IN AN ULTRA WIDEBAND SOURCE (UWBS) INTERFERENCE

by

Yeu, Eng-Kiong

September, 1996

Thesis Advisor: Tri T. Ha

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 1996	Master's Thesis

4. TITLE AND SUBTITLE
ERROR PROBABILITIES OF SPREAD SPECTRUM SYSTEMS IN AN ULTRA WIDEBAND SOURCE (UWBS) INTERFERENCE

5. FUNDING NUMBERS

6. AUTHOR(S) Yeu, Eng-Kiong

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis investigates the potential jamming capabilities of an ultra wideband source (UWBS) jammer on the direct sequence (DS) and frequency hopping (FH) spread spectrum systems using the Advanced Communication Link Analysis and Design (ACOLADE) tool. Error probabilities in both the additive white Gaussian noise channel and the Rayleigh fading channel are obtained with and without convolutional coding. Comparison is made to examine whether an UWBS jamming is more effective on the DS or FH systems given the same jammer power. The thesis also presents how several jammer parameters could be varied in order to inflict more harm on the communication systems.

14. SUBJECT TERMS. Spread Spectrum System, Ultra Wideband Source, Coding			15. NUMBER OF PAGES 72
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

ERROR PROBABILITIES OF SPREAD SPECTRUM SYSTEMS IN ULTRA WIDEBAND SOURCE (UWBS) INTERFERENCE

Yeu, Eng-Kiong
Defense Technology Group, Singapore
B.Eng., National University of Singapore, Singapore, 1991

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1996

Author: [redacted]

Yeu, Eng-Kiong

Approved by: [redacted]

Tri T. Ha, Thesis Advisor

[redacted]

Vicente Garcia, Co-Advisor

[redacted]

Herschel H. Loomis, Jr., Chairman
Department of Electrical and Computer Engineering

ABSTRACT

This thesis investigates the potential jamming capabilities of an ultra wideband source (UWBS) jammer on the direct sequence (DS) and frequency hopping (FH) spread spectrum systems using the Advanced Communication Link Analysis and Design (ACOLADE) tool. Error probabilities in both the additive white Gaussian noise channel and the Rayleigh fading channel are obtained with and without convolutional coding. Comparison is made to examine whether an UWBS jamming is more effective on the DS or FH systems given the same jammer power. The thesis also presents how several jammer parameters could be varied in order to inflict more harm on the communication systems.

TABLE OF CONTENTS

I. INTRODUCTION

A. A SPREAD SPECTRUM OVERVIEW

The investigation of spread spectrum techniques was motivated primarily by the desire to achieve highly jam-resistant communication systems. A spread spectrum system is capable of combating or suppressing the detrimental effects of interference due to jamming, interference arising from other users of the channel, and self-interference due to multipath propagation. Low probability of detection (LPD) can be achieved which makes signal detection difficult for an unintended listener in the presence of background noise. It is also capable of employing low probability of intercept (LPI) methods to achieve message privacy in a hostile environment. In this report, we will focus on two major spread spectrum techniques, namely, the direct sequence spread spectrum (DS/SS) and the frequency hopping spread spectrum (FH/SS).

B. THE ULTRA WIDEBAND SOURCE (UWBS) JAMMER

The goals of a jammer are to deny reliable communications to an adversary and to accomplish this at a minimum cost. Jamming can be defined as the deliberate emission of electromagnetic (EM) radiation to reduce and prevent the use of a portion of EM spectrum by other parties. The recent development of the ultra wideband source (UWBS) device and its availability pose a serious threat to current communication systems. The UWBS device is capable of emitting pulses of duration in the nanoseconds and with pulse repetition frequencies (PRFs) up to 50 kHz. It is flexible to vary its pulse duration and duty cycle

to concentrate a high amount of energy in a particular bandwidth. As the UWBS device is compact in size, the jammer is also suitable for mobile deployments. For instance, the smallest UWBS device has a size of 4" x 3" x 1" and can emit a pulse train with PRF up to 50 kHz, a pulse duration of about 5 ns, and a peak power of 2 kW. A more powerful UWBS device, weighing about 1500 lbs with a size of 63" x 63" x 34", can generate as much as 1 GW of peak power with an electronically steerable array.

C. MAIN RESEARCH THRUST AND APPROACH

As the existence of the UWBS device is fairly recent, it is of great interest to analyze its impact on current communication systems. The main thrust of this research is thus to investigate the potential jamming capability of an UWBS interference on both the DS/SS and FH/SS systems with and without forward error correction. A simulation tool named the Advanced Communication Link Analysis and Design (ACOLADE) software was used to design and implement the Monte Carlo simulations for the communication systems under test. The unique simulation environment was based on the concept of models which organizes subsystems within the digital topology. It simulates the performance of each subsystem by synthesizing the time-domain analog or digital data and passing it between the various building blocks where it is in turn processed by each model. At the output of the digital link, the demodulated and decoded data stream is compared with the digital source to produce the desired error probabilities.

D. REPORT ORGANIZATION

In the next chapter, the ACOLADE simulation environment is described and the steps taken to formulate the problems are examined together with some theories involved. Chapter III explains the scenario of each simulation run and the respective assumptions. Chapter IV discusses the simulation results in detail and provides a good insight of the UWBS jamming effects on the spread spectrum systems. Chapter V concludes the findings and results of this research.

II. ACOLADE SIMULATION ENVIRONMENT

A. INTRODUCTION

The Advanced Communication Link Analysis and Design (ACOLADE) is a software tool that enables an user to design and implement a Monte Carlo simulation for a communication system under test. The simulation is based on the concept of models whereby each model corresponds to a particular component in a communication system such as a transmitter, a channel or a receiver [Refs. 10 and 11]. The software has prepackaged models that are grouped according to classes e.g., "Digital Source," "Gaussian Noise Source" and "Random Number Generator" are some of the models that are categorized under the class of "Sources." These models are written in the C language and because of the inherent powerful capability of this programming language, the users enjoy a high degree of flexibility to modify the prepackaged models or even to create their own application-specific models to suit a particular communication scenario. As an example, Appendix A shows the listing of "Direct Sequence Spread Spectrum" model, written in the C language, which was modified to meet our desired specifications and also to ensure consistent interfaces when it is connected to other models in the program.

B. PROGRAM ARCHITECTURE

The simulation environment consists of several key elements, namely, the Executive Command, the Graphical User Interface, the Model and its Interface File, the Simulation Topology File, and the Global Parameter Facility.

The Executive Command, as the name implies, is the command and control center which drives all simulation routines. It is responsible for maintaining the parameter lists of all the models used and for storing/retrieving information of a particular design to and from the respective files which contain all relevant interconnection information and parameters. Its tasks also include scheduling the order in which the models are to be invoked, providing for the storage of data and variables, as well as facilitating the transfer of data between models. On the other hand, the Graphical User Interface provides the users a friendly and intuitive environment to specify the models and their interconnections.

The Model and its Interface File is another key element which defines the model parameters, the model ports and the data types (e.g., digital data, real data and complex data), and the iteration parameters to step or reset a simulation routine. In order to keep track of the models used and to maintain topology records during a simulation, the Simulation Topology File commands two important data files, namely, the INT file and the NET file. The INT file contains information of every model that the Executive Command requires to schedule it for inclusion in a particular routine. It is also the function of this file to define the external characteristics such as the number and the type of input/output ports of a model. Each time a simulation is constructed using the Graphical User Interface, a NET file is created which contains all necessary information the Executive Command requires to execute the respective operation. As an example, Appendix B gives the listing of the NET file created in order to obtain the error probabilities of the DS/SS system in an UWBS interference in the additive white Gaussian noise channel.

The Global Parameter Facility organizes the parameter tables which are associated with the top-level network nodes or sub-nodes in order to set, read and remove entries from the tables. It is referenced within the NET file and is designed to effect the passing of parameters throughout the network tree and to allow the executing modules to create, update and remove these parameters.

C. PROBLEM FORMULATION

In order to obtain the error probabilities of the spread spectrum systems in an UWBS interference, it is first necessary to formulate the problems in the ACOLADE simulation environment. The various system configurations employed in the simulation are described in the following subsections.

1. Direct Sequence Spread Spectrum (DS/SS)

At the transmitter of the DS/SS system, the message signal m(t) is first spread by the pseudo-noise (PN) signal c(t) using multiplication. The resulting signal m(t)c(t) is then modulated onto a carrier using the coherent binary phase shift keying (BPSK) modulation which yields the DS/SS-BPSK signal given by

$$s(t) = A\, m(t)c(t)\cos(2\pi f_c t + \theta) \tag{2.1}$$

where A is the amplitude, f_c is the carrier frequency, θ is the carrier phase. In the simulations, m(t) is assumed to have a bit rate of 20 kbps. c(t) is implemented using a m-sequence of length N = 1023 which corresponds to a processing gain of about 30 dB.

At the receiver, the received signal is first despread by the same chipping signal c(t). An estimate of the message m(t) is then recovered using a conventional BPSK demodulator. In the presence of barrage noise interference in the AWGN channel, the received signal is given by

$$r(t) = s(t) + n_o(t) + n_j(t) \tag{2.2}$$

where $n_o(t)$ is the AWGN process with two-sided power spectral density of $N_o/2$ and $n_j(t)$ represents an interfering signal with two-sided power spectral density of $N_j/2$ within the null-to-null bandwidth of the DS/SS-BPSK signal. The bit error probability is thus given by [Ref. 3]

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_o + N_j}}\right) \tag{2.3}$$

where Q(x) is the Q function and E_b is the energy per bit.

2. Frequency Hopping Spread Spectrum (FH/SS)

At the transmitter of the FH/SS system, the transmitted frequency is produced by a frequency synthesizer whose output frequency changes every T_h seconds according to the values of j bits, i.e., the hopping pattern is determined by j-1 bits from the PN sequence generator and 1 bit from the message signal. As j bits give a total of 2^j combinations, we can therefore have 2^j possible frequencies using the binary frequency shift keying (BFSK) modulation. For orthogonal signaling, it requires that the minimum frequency separation be $1/T_h$ which results in a spread spectrum bandwidth of $2^j/T_h$ Hz. In the simulations, a

fast frequency hopping system was used with two hops per bit and the value of j was assumed to be ten. The signal at the frequency synthesizer during the k-th hop can thus be written as

$$s(t) = A \cos(2\pi(f_o + i_k\Delta f + d_k\Delta f)t + \theta_k) \tag{2.4}$$

where $i_k \in \{1, 3, \ldots, M\text{-}1\}$ is an odd integer, $d_k \in \{0, 1\}$ is the value of the data bit during $kT_h < t < (k+1)T_h$. The value of i_k is determined by the j-1 bits from the PN sequence generator. The transmitted frequency is $f_o + i_k\Delta f$ when $d_k = 0$ and is $f_o + i_k\Delta f + \Delta f$ when $d_k = 1$, and θ_k represents the carrier phase which has an uniform distribution over the interval $[0, 2\pi]$.

At the receiver, the local PN sequence generator produces a PN sequence in synchronism with the incoming one. During the k-th hop, the output of the frequency synthesizer is given by

$$g(t) = \cos(2\pi(f_o + (i_k - 1)\Delta f)t + \theta_k), \; for \; kT_h < t < (k+1)T_h \tag{2.5}$$

where f_o is the center carrier frequency.

Ignoring the high frequency terms, the product of s(t)g(t) which represents the input to a conventional BFSK demodulator can be simplified as

$$s_1(t) = \frac{A}{2} \cos(2\pi f_d t + \theta_k) \tag{2.6}$$

which contains frequency f_d of either Δf or $2\Delta f$ depending on $d_k = 0$ or 1 respectively. The demodulator then detects the frequency contained in each bit duration and produces a binary output of "0" or "1".

3. Ultra Wideband Source (UWBS) Signal

The UWBS signal is modeled as a coherent pulse train with a pulse width, T and a pulse repetition frequency, f_p. In the simulations, it was assumed that the UWBS device was capable of emitting pulses with T = 5 ns and f_p = 40 kHz. The relationship between the pulse width and the bandwidth of the pulse wave can be given by [Ref. 9]

$$B \approx \frac{1}{T} \tag{2.7}$$

On the other hand, the relationship between the average power, P_{ave} and the peak pulse envelope power, P_p of the UWBS signal is given by

$$P_{ave} = P_p \frac{T}{T_p} \tag{2.8}$$

where T_p is the reciprocal of f_p.

4. Convolutional Code

A convolutional code can be generated by passing the information sequence to be transmitted through a linear finite-state shift register. The shift register consists of K stages and n linear algebraic function generators. The encoder has k inputs and n outputs. The code rate, R is defined as k/n and K is known as the constraint length of the code. In the simulations, R = ½ and K = 7 are used.

The optimum decoding of a convolutional code employs the Viterbi algorithm which involves a search through the trellis for the most probable sequence [Refs. 7 and 8]. It was implemented as an iterative algorithm by correlating the received sequences with all the possible transmitted sequences until the one with the best 'fit' is arrived at. In the simulations, a hard decision decoding is used as opposed to a soft decision decoding. The metrics in this case are the Hamming distances between the $2^{k(K-1)}$ surviving sequence and the received sequence at each node of the trellis.

III. SIMULATION RESULTS

A. ERROR PROBABILITIES OF THE DS/SS-BPSK SYSTEM

Figure 1 shows the bit error probabilities for the DS/SS-BPSK system in the AWGN channel with an UWBS interference. Three jamming-to-signal ratios (J/S) of 10 dB, 20 dB and 30 dB were considered with a PRF and a pulse width, T, of 40 kHz and 5 ns respectively. Figure 2 examines the effect on the error probability due to variations in the PRF. Three PRF values of 10 kHz, 20 kHz and 40 kHz were considered with J/S and T of 20 dB and 5 ns respectively. On the other hand, Figure 3 examines the effect on the error probability due to variations in T. Three T values of 5 ns, 10 ns and 20 ns were considered with J/S and PRF of 20 dB and 40 kHz respectively.

Figures 4 to 6 focus on the error probabilities of the DS/SS-BPSK system in the Rayleigh fading channel with an UWBS interference. To elaborate, Figure 4 presents the bit error probabilities for the DS/SS-BPSK system in the Rayleigh fading channel with an UWBS interference where three J/S ratios of 6 dB, 16 dB and 26 dB were considered. The values of PRF and T were assumed to be 40 kHz and 5 ns respectively. Likewise as in Figure 2, Figure 5 examines the effect on the error probability due to variations in the PRF. Two PRF values of 10 kHz and 40 kHz were considered with J/S and T of 16 dB and 5 ns respectively. Similarly as in Figure 3, Figure 6 examines the effect on the error probability due to variations in T. Two values of T of 5 ns and 20 ns were considered with J/S and PRF of 16 dB and 40 kHz respectively.

Having examined the uncoded performance of the DS/SS-BPSK system in Figures 1 to 6, Figure 7 demonstrates the coding gain achieved with the convolutional coding of rate, R = ½ and a constraint length, K = 7. Two J/S ratios of 10 dB and 20 dB were considered in the AWGN channel with PRF and T assumed to be 40 kHz and 5 ns respectively. The respective uncoded error probabilities were also shown in the same Figure for comparison purpose. Figures 8 and 9 compare the coded and uncoded performances of the DS/SS-BPSK system in the Rayleigh fading channel. The only difference in these two Figures lies in the J/S ratio used. J/S ratios of 6 dB and 16 dB were considered in these Figures respectively. The same convolutional code of R = ½ and K = 7 was used with PRF and T of 40 kHz and 5 ns respectively.

B. ERROR PROBABILITIES OF THE FH/SS-BFSK SYSTEM

The descriptions in the preceding section are pertaining to the DS/SS-BPSK system. The focus of this section, on the other hand, is on the error probabilities of the FH/SS-BFSK system with an UWBS interference.

Figure 10 shows the bit error probabilities for the FH/SS-BFSK system in the AWGN channel where three J/S ratios of 10 dB, 20 dB and 30 dB were considered. The values of PRF and T were assumed to be 40 kHz and 5 ns respectively. Figure 11 presents the effect on the error probability due to variations in the PRF. Two PRF values of 10 kHz and 40 kHz were considered with J/S and T of 20 dB and 5 ns respectively.

On the other hand, Figure 12 looks at the effect on the error probability due to variations in T. Two values of T of 5 ns and 20 ns were considered with J/S and PRF of 20 dB and 40 kHz respectively.

Figures 13 to 15 focus on the error probabilities of the FH/SS-BFSK system in the Rayleigh fading channel with an UWBS interference. To elaborate, Figure 13 presents the bit error probabilities for the FH/SS-BFSK system in the Rayleigh fading channel with an UWBS interference where three J/S ratios of 6 dB, 16 dB and 26 dB were considered. The values of PRF and T were assumed to be 40 kHz and 5 ns respectively. Figure 14 examines the effect on the error probability due to variations in the PRF. Two PRF values of 10 kHz and 40 kHz were considered with J/S and T of 16 dB and 5 ns respectively. Figure 15 examines the effect on the error probability due to variations in T. Two values of T of 5 ns and 20 ns were considered with J/S and PRF of 16 dB and 40 kHz respectively.

Having examined the uncoded performance of the FH/SS-BFSK system in Figures 10 to 15, it was also of interest to investigate the coded performance of the same system with an UWBS interference. Thus as shown in Figure 16, the coding gain achieved with the convolutional coding was presented. Two J/S ratios of 10 dB and 20 dB were considered in the AWGN channel with PRF and T of 40 kHz and 5 ns respectively. The respective uncoded error probabilities were also shown in the same Figure for comparison purpose.

Lastly, Figures 17 and 18 compare the coded and uncoded performances of the FH/SS-BFSK system in the Rayleigh fading channel. The only difference in these two Figures lies in the J/S ratio used. J/S ratios of 6 dB and 16 dB were considered in these Figures respectively. The same convolutional code of R = ½ and K = 7 was used with PRF and T of 40 kHz and 5 ns respectively.

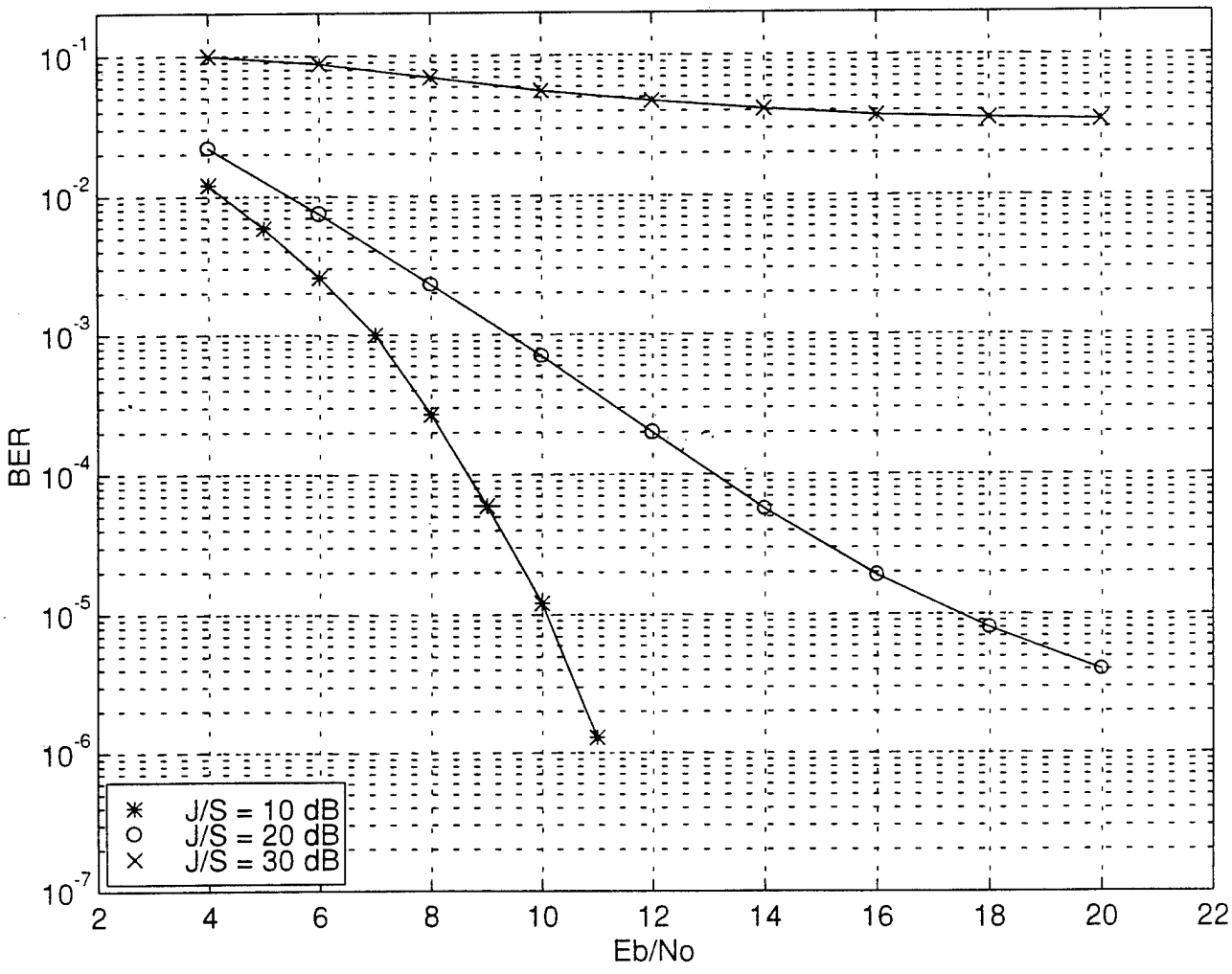

Figure 1. DS/SS-BPSK with UWBS PRF = 40 kHz and T = 5 ns in AWGN Channel

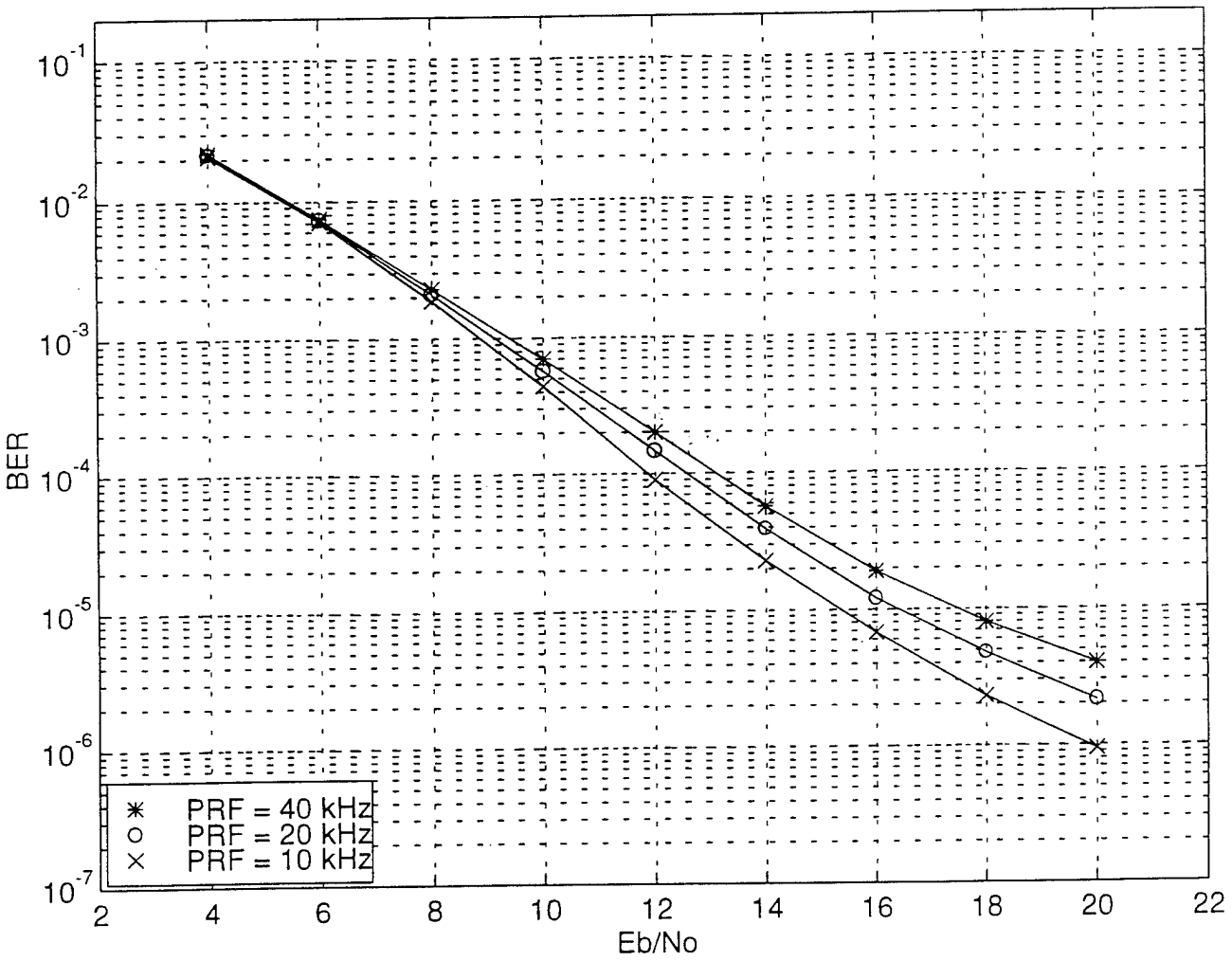

Figure 2. DS/SS-BPSK with UWBS J/S = 20 dB and T = 5 ns in AWGN Channel

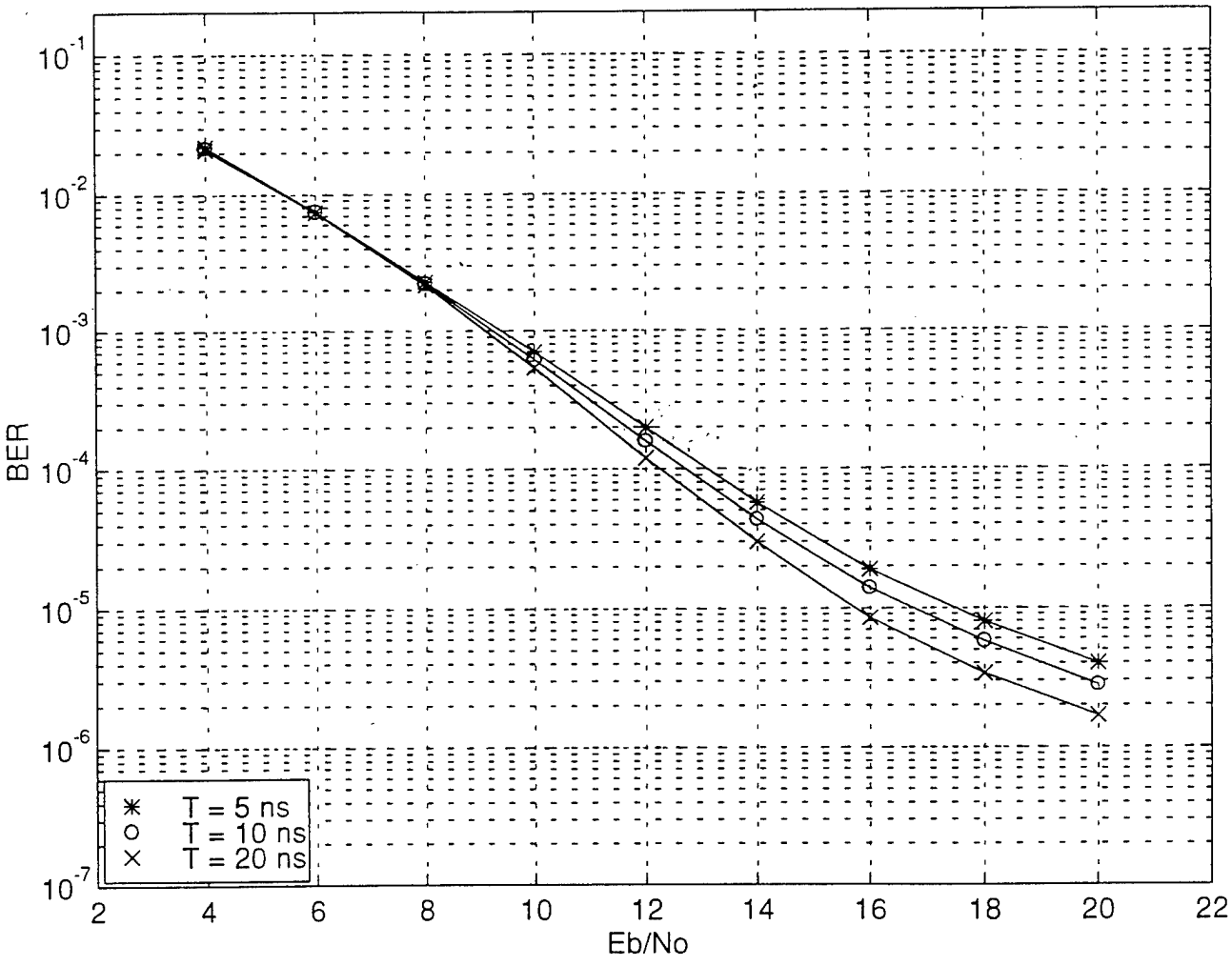

Figure 3. DS/SS-BPSK with UWBS J/S = 20 dB and PRF = 40 kHz in AWGN Channel

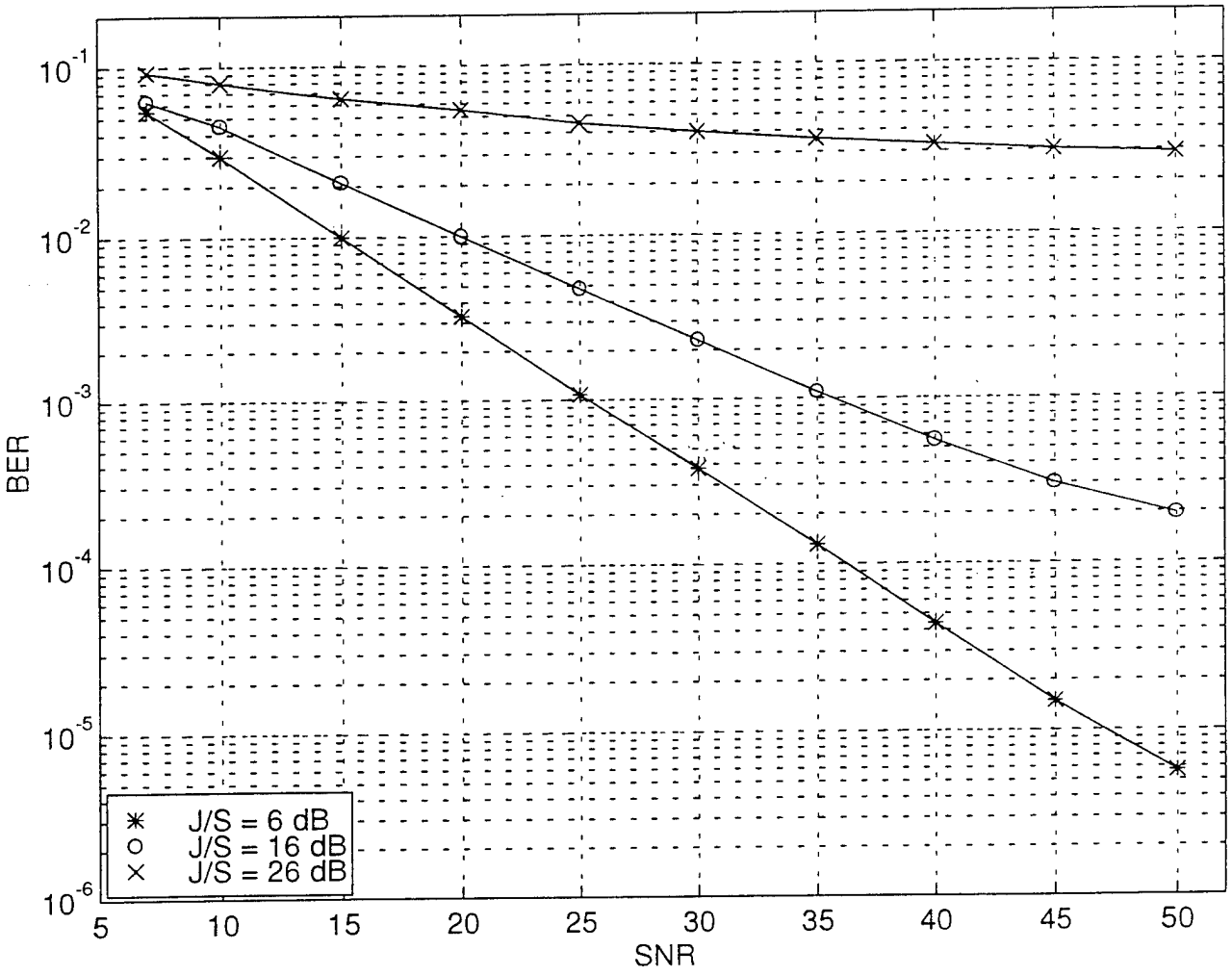

Figure 4. DS/SS-BPSK with UWBS PRF = 40 kHz and T = 5 ns in Rayleigh Channel

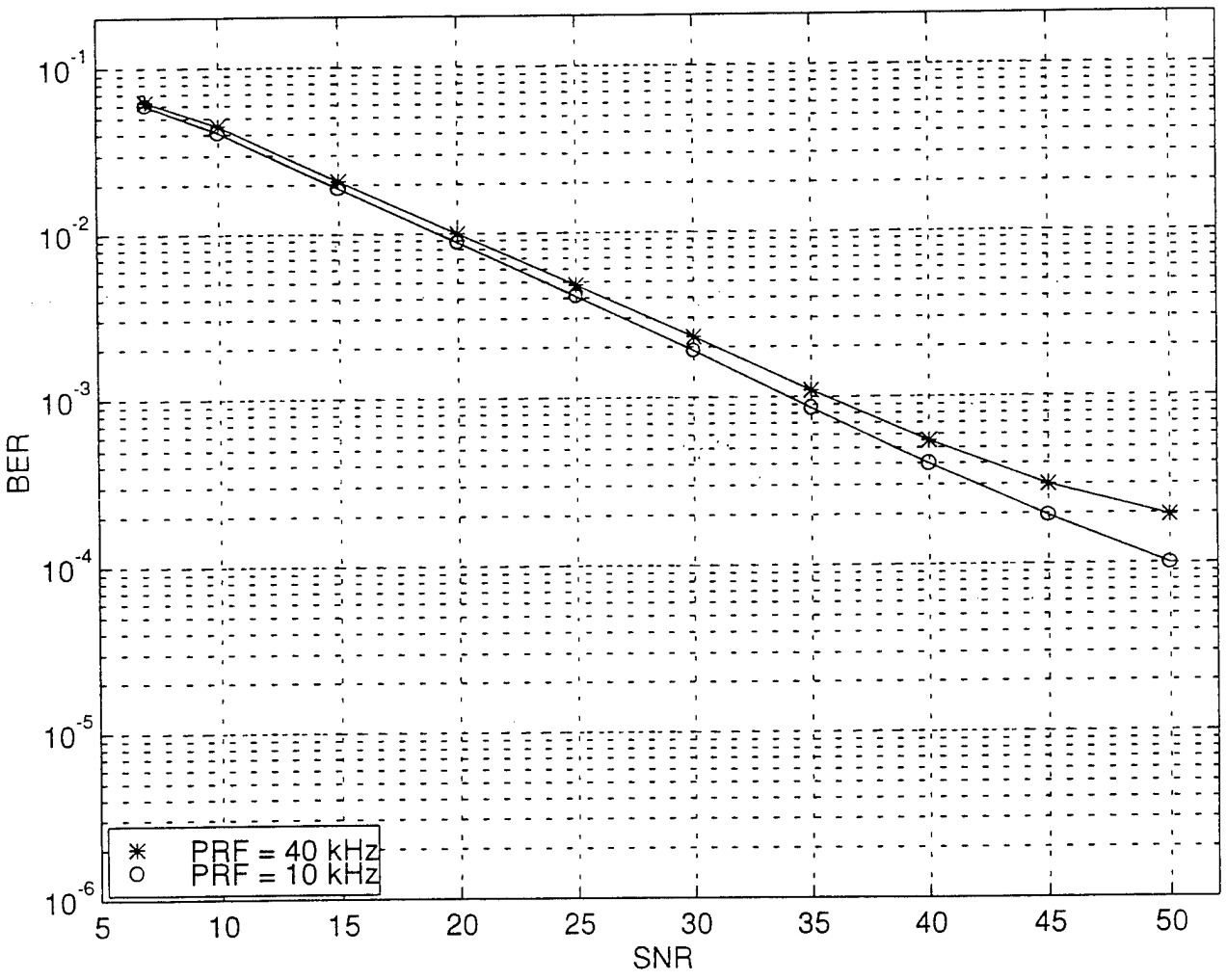

Figure 5. DS/SS-BPSK with UWBS J/S = 16 dB and T = 5 ns in Rayleigh Channel

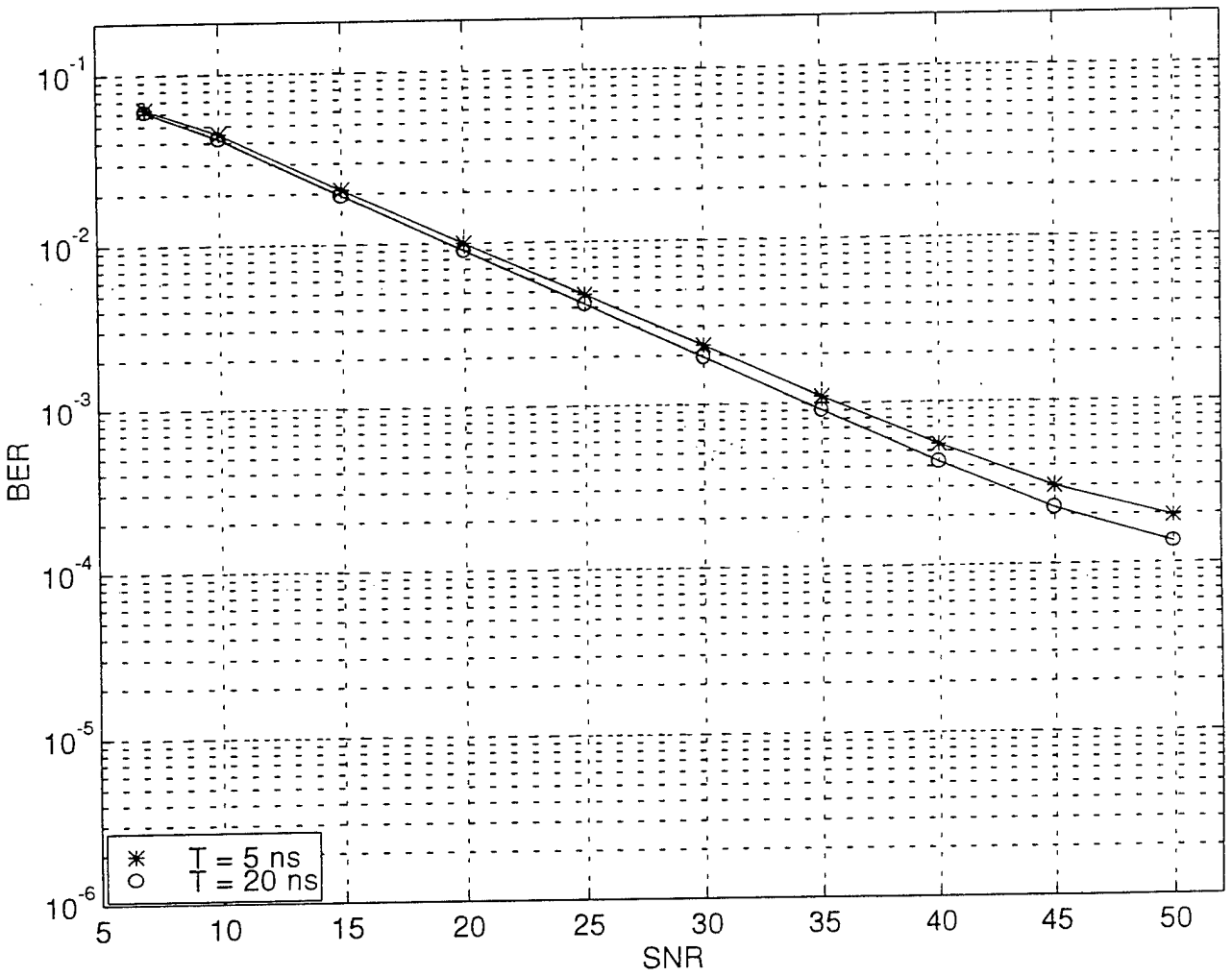

Figure 6. DS/SS-BPSK with UWBS J/S = 16 dB and PRF = 40 kHz in Rayleigh Channel

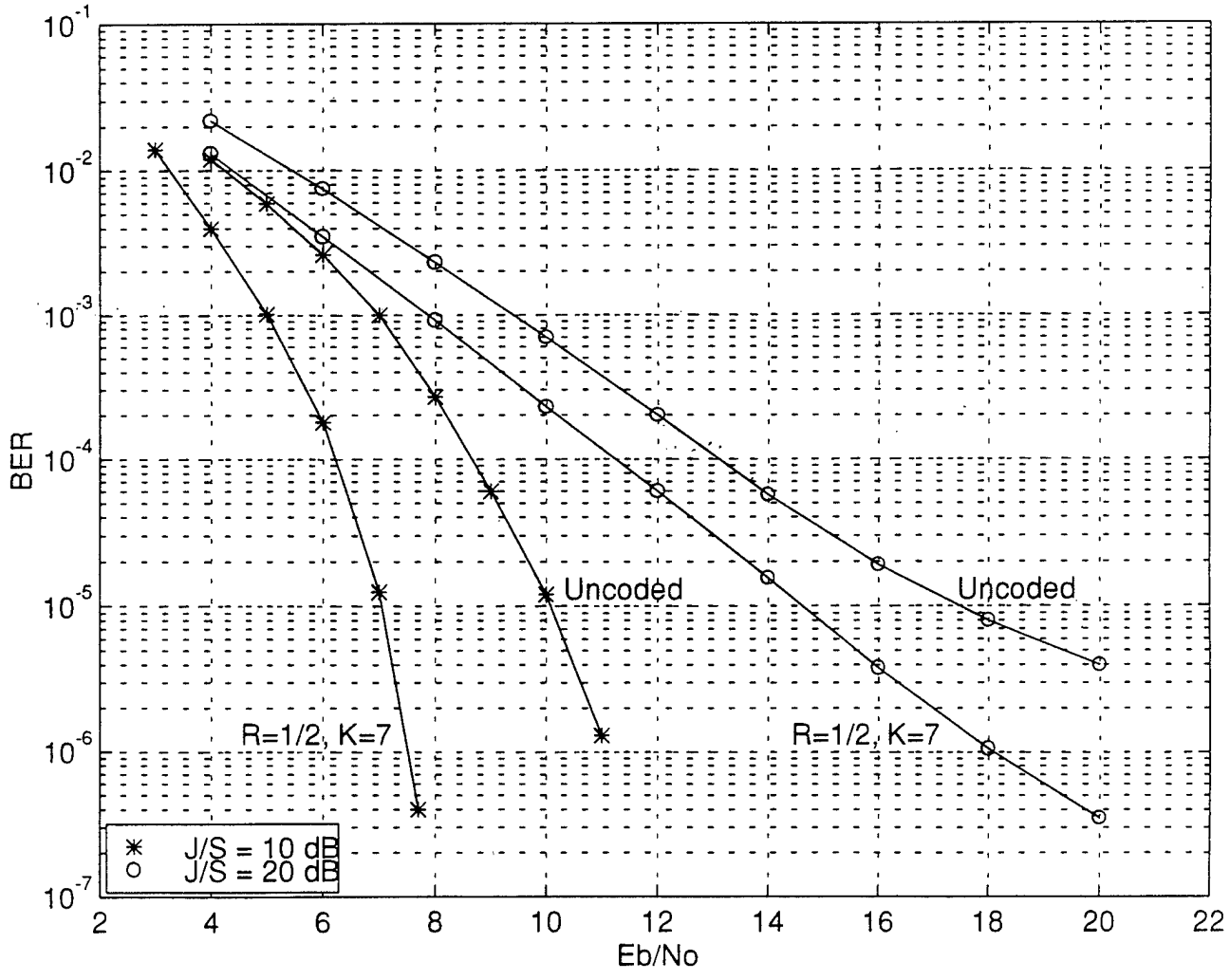

Figure 7. Coded DS/SS-BPSK with UWBS PRF = 40 kHz and T = 5 ns in AWGN Channel

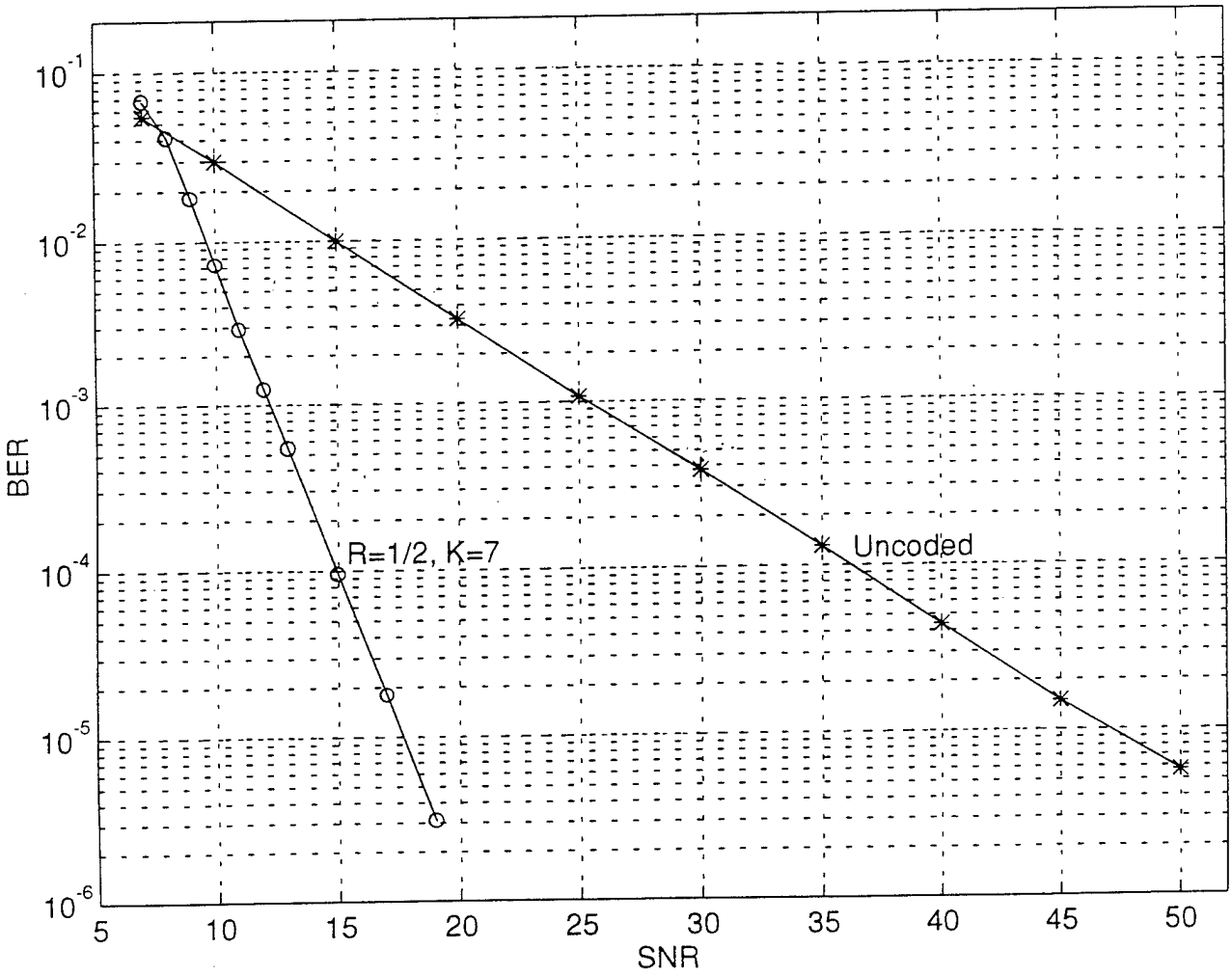

Figure 8. Coded DS/SS-BPSK with UWBS J/S = 6 dB, PRF = 40 kHz and T = 5 ns in Rayleigh Channel

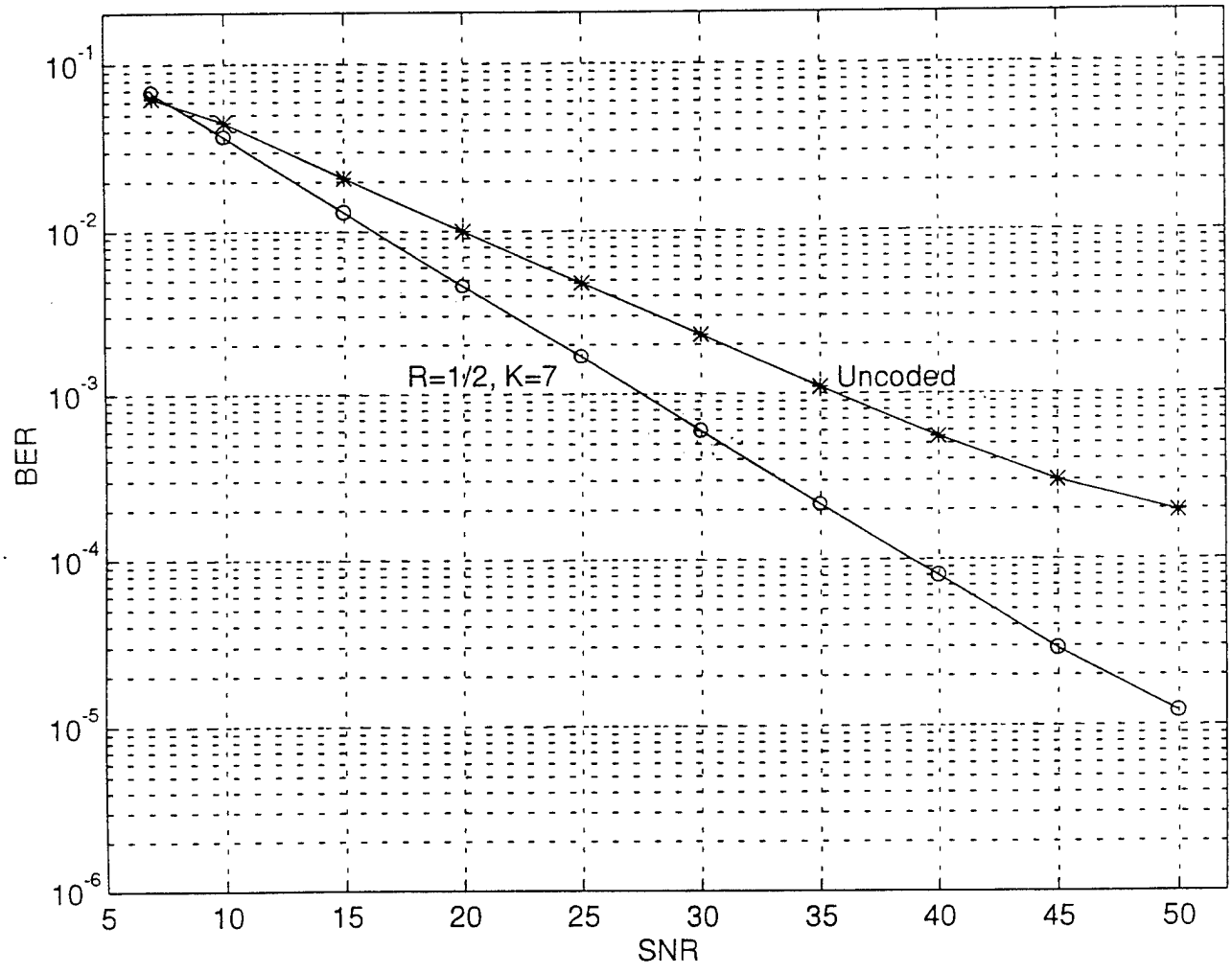

Figure 9. Coded DS/SS-BPSK with UWBS J/S = 16 dB, PRF = 40 kHz and T = 5 ns in Rayleigh Channel

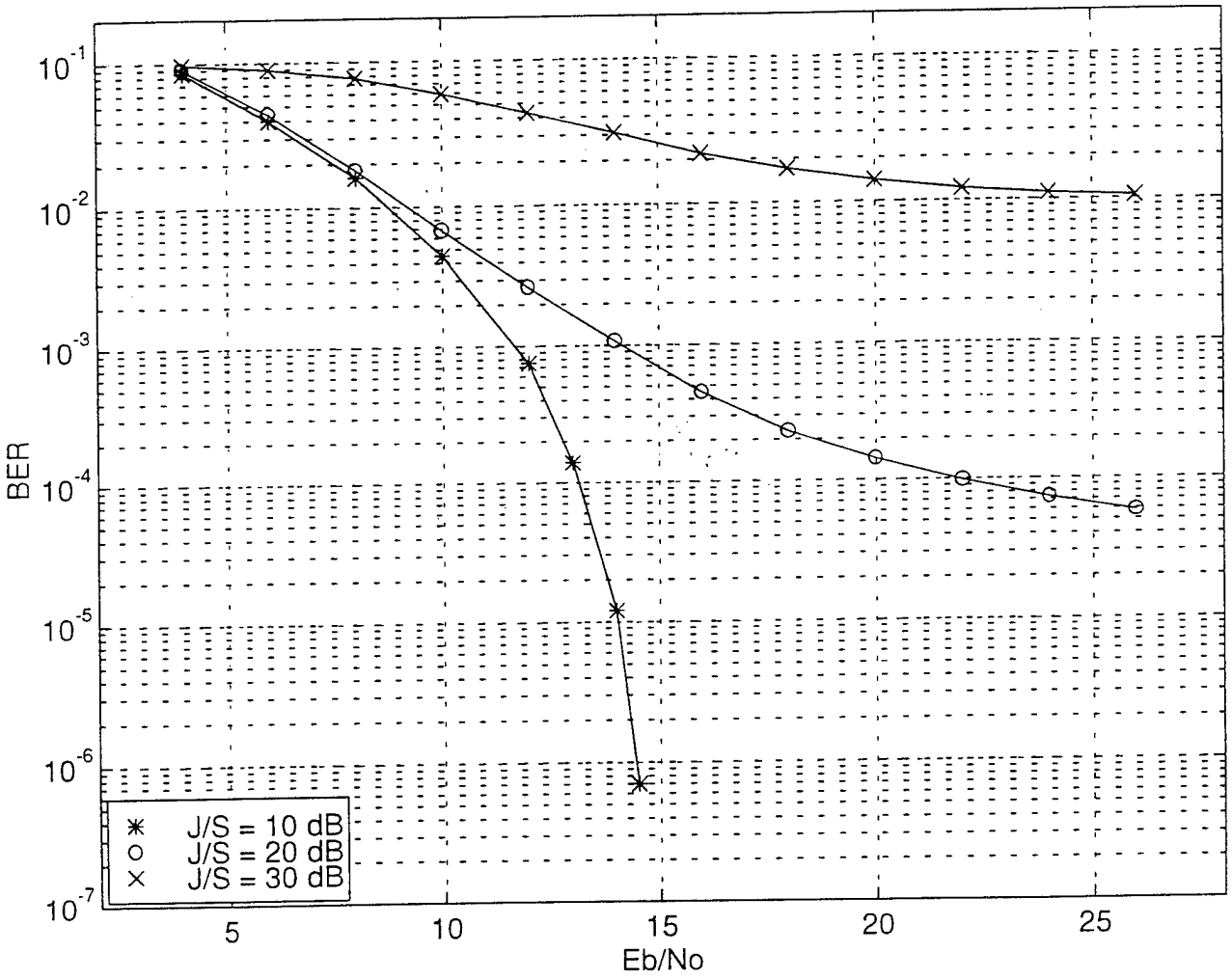

Figure 10. FH/SS-BFSK with UWBS PRF = 40 kHz and T = 5 ns in AWGN Channel

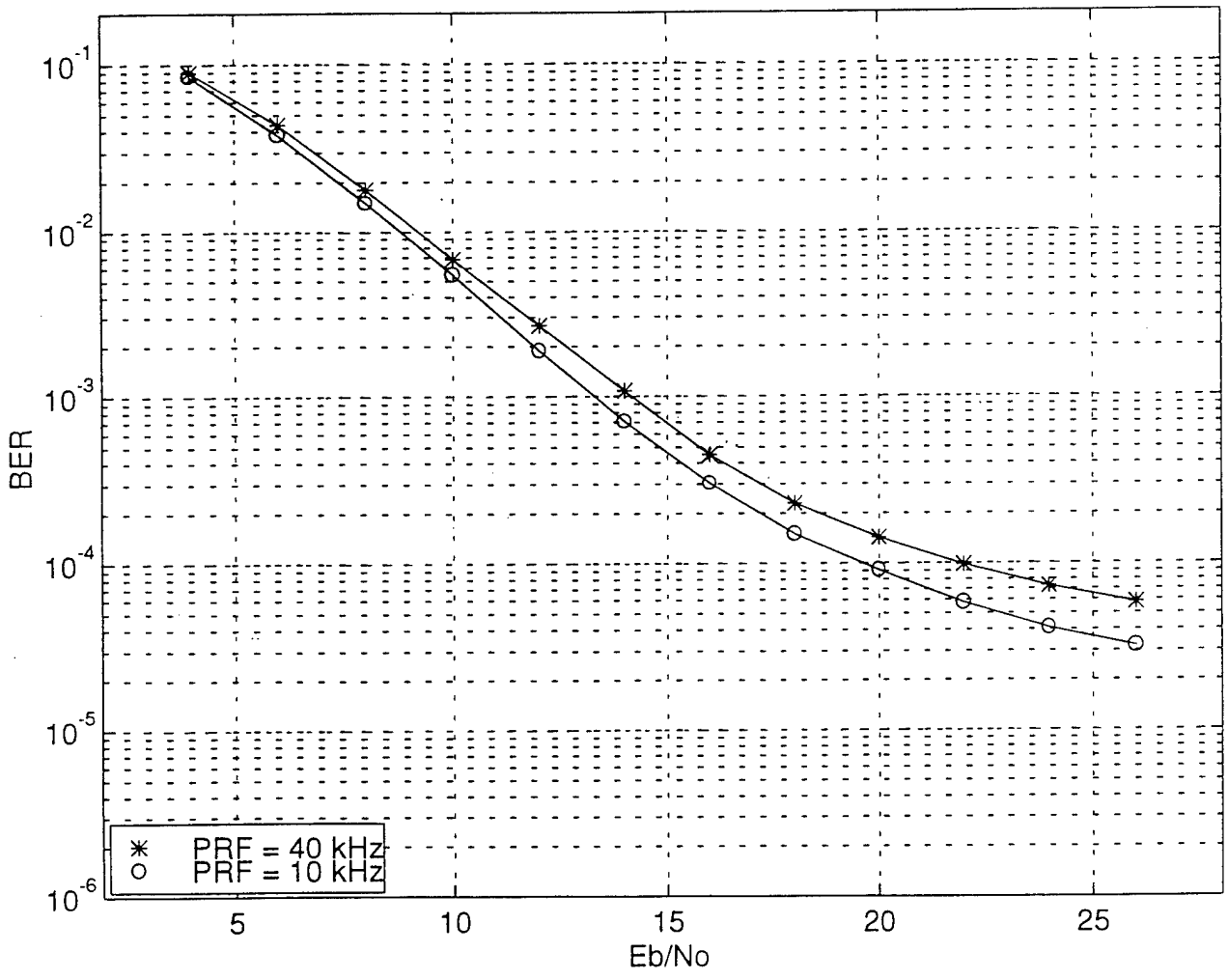

Figure 11. FH/SS-BFSK with UWBS J/S = 20 dB and T = 5 ns in AWGN Channel

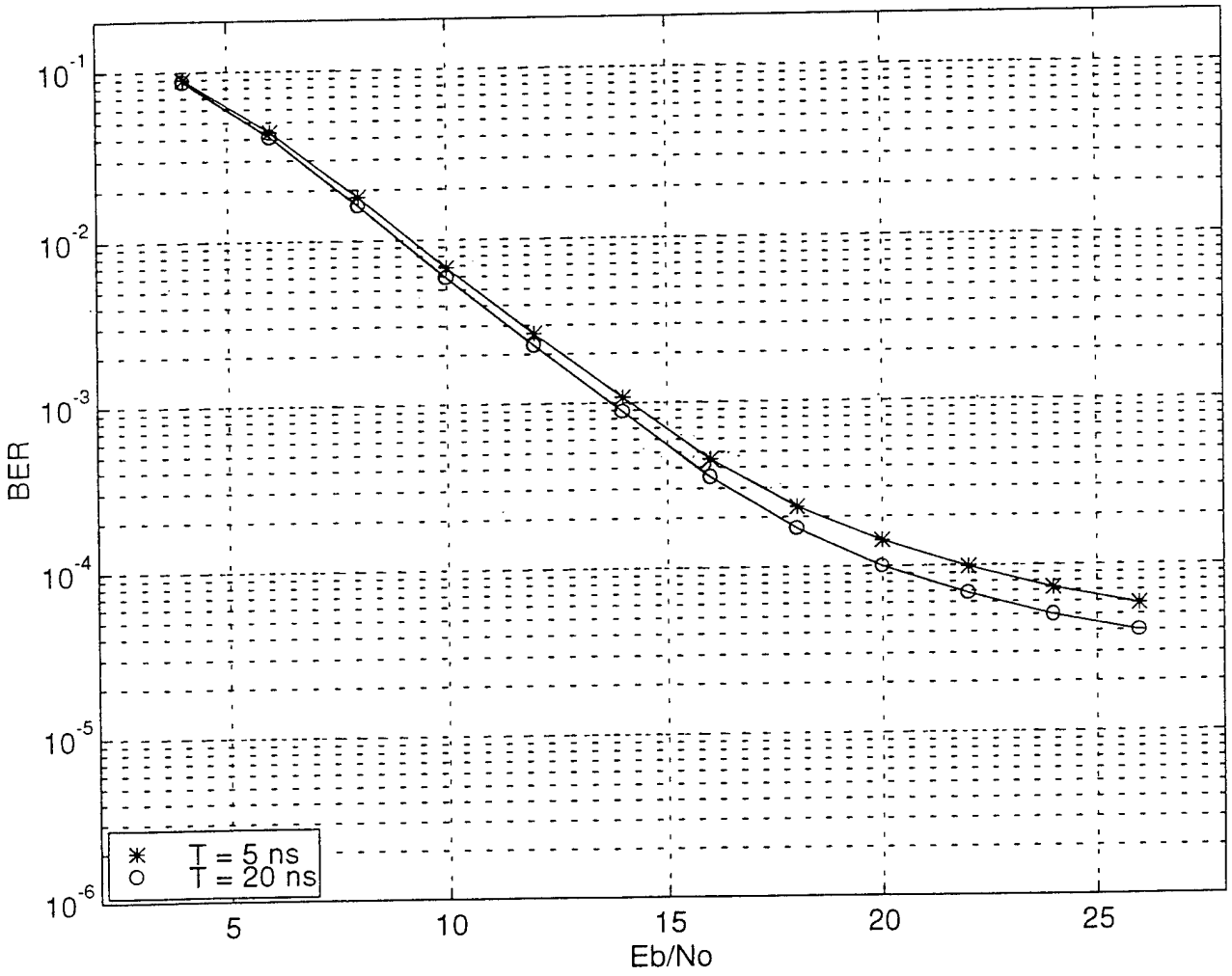

Figure 12. FH/SS-BFSK with UWBS J/S = 20 dB and PRF = 40 kHz in AWGN Channel

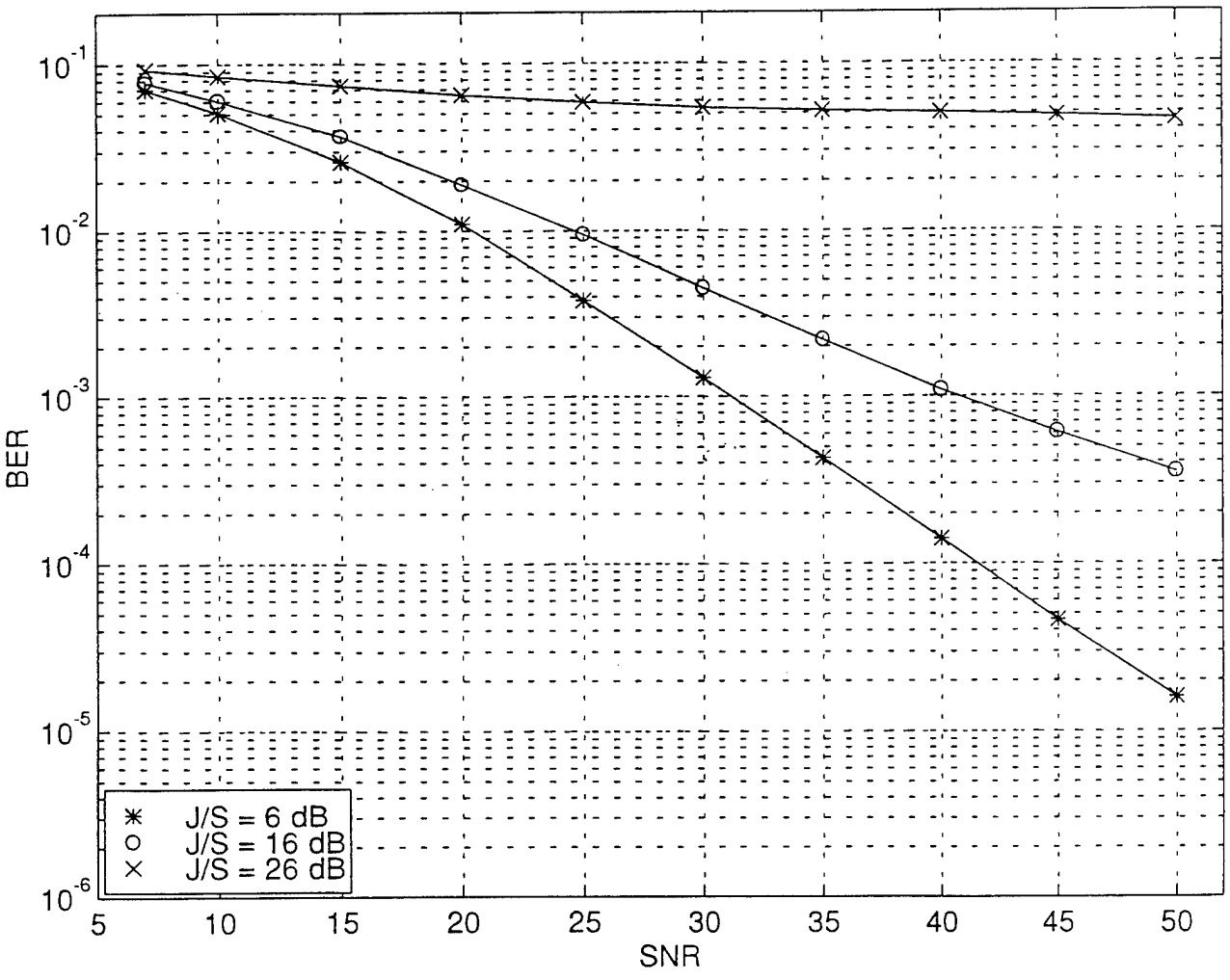

Figure 13. FH/SS-BFSK with UWBS PRF = 40 kHz and T = 5 ns in Rayleigh Channel

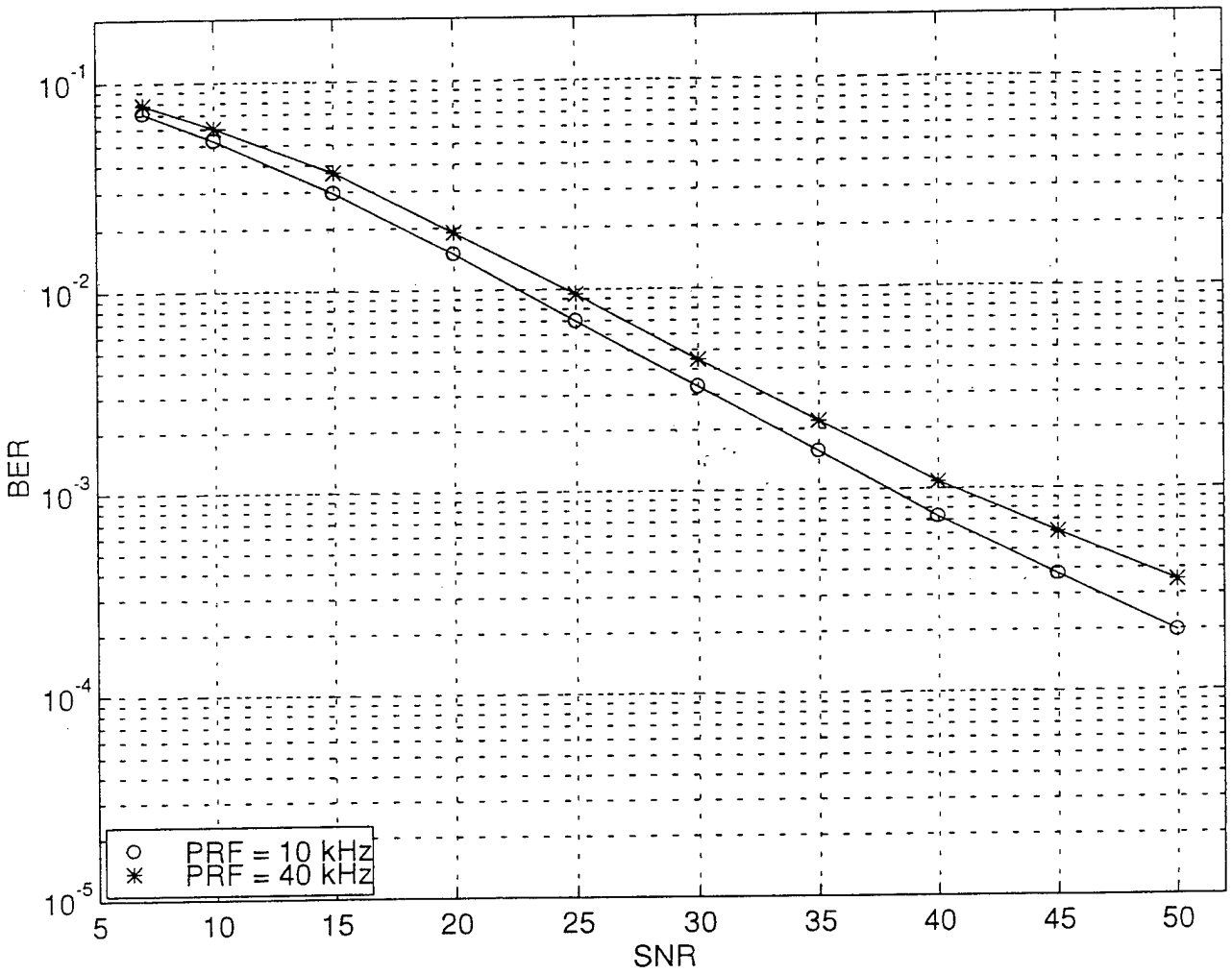

Figure 14. FH/SS-BFSK with UWBS J/S = 16 dB and T = 5 ns in Rayleigh Channel

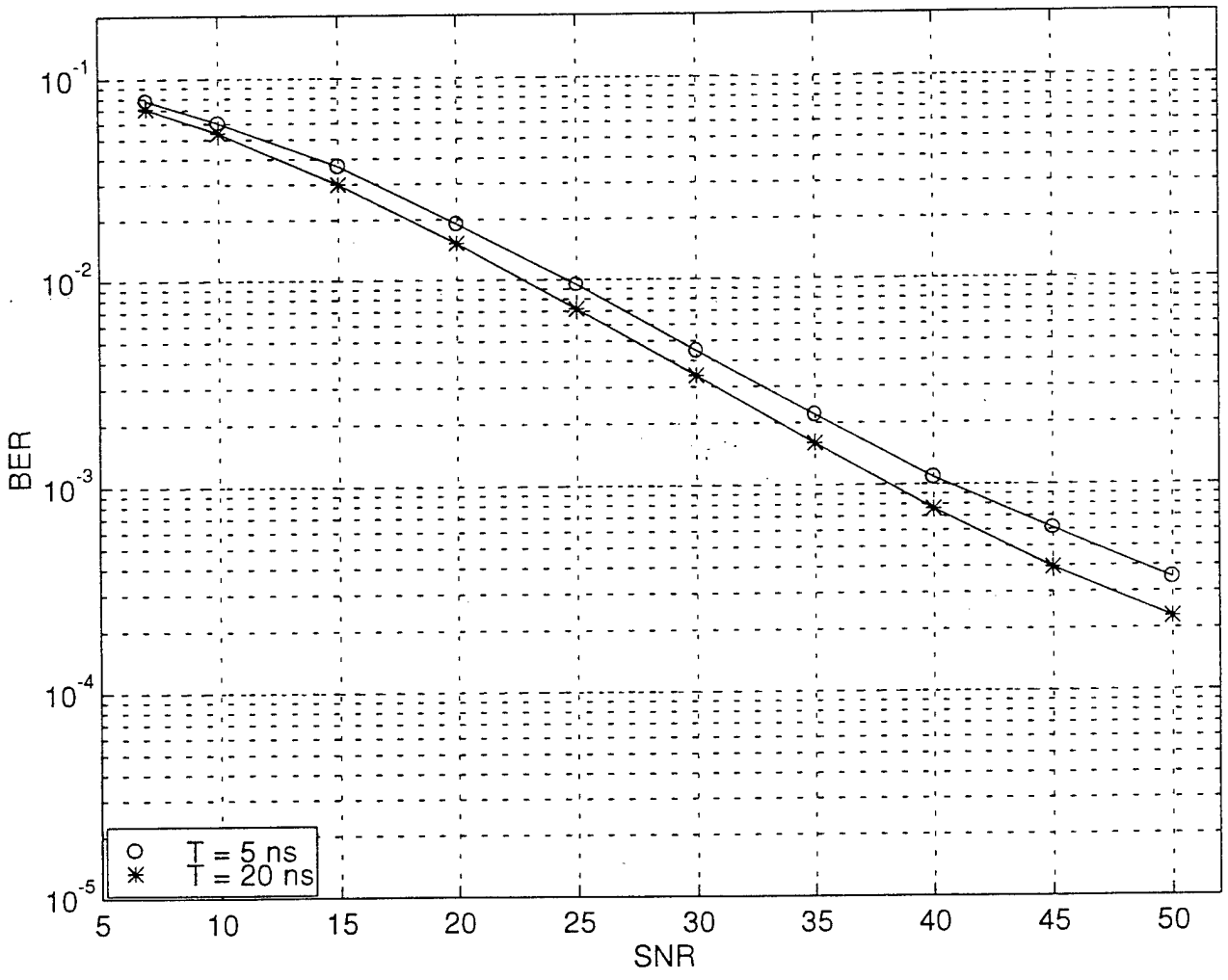

Figure 15. FH/SS-BFSK with UWBS J/S = 16 dB and PRF = 40 kHz in Rayleigh Channel

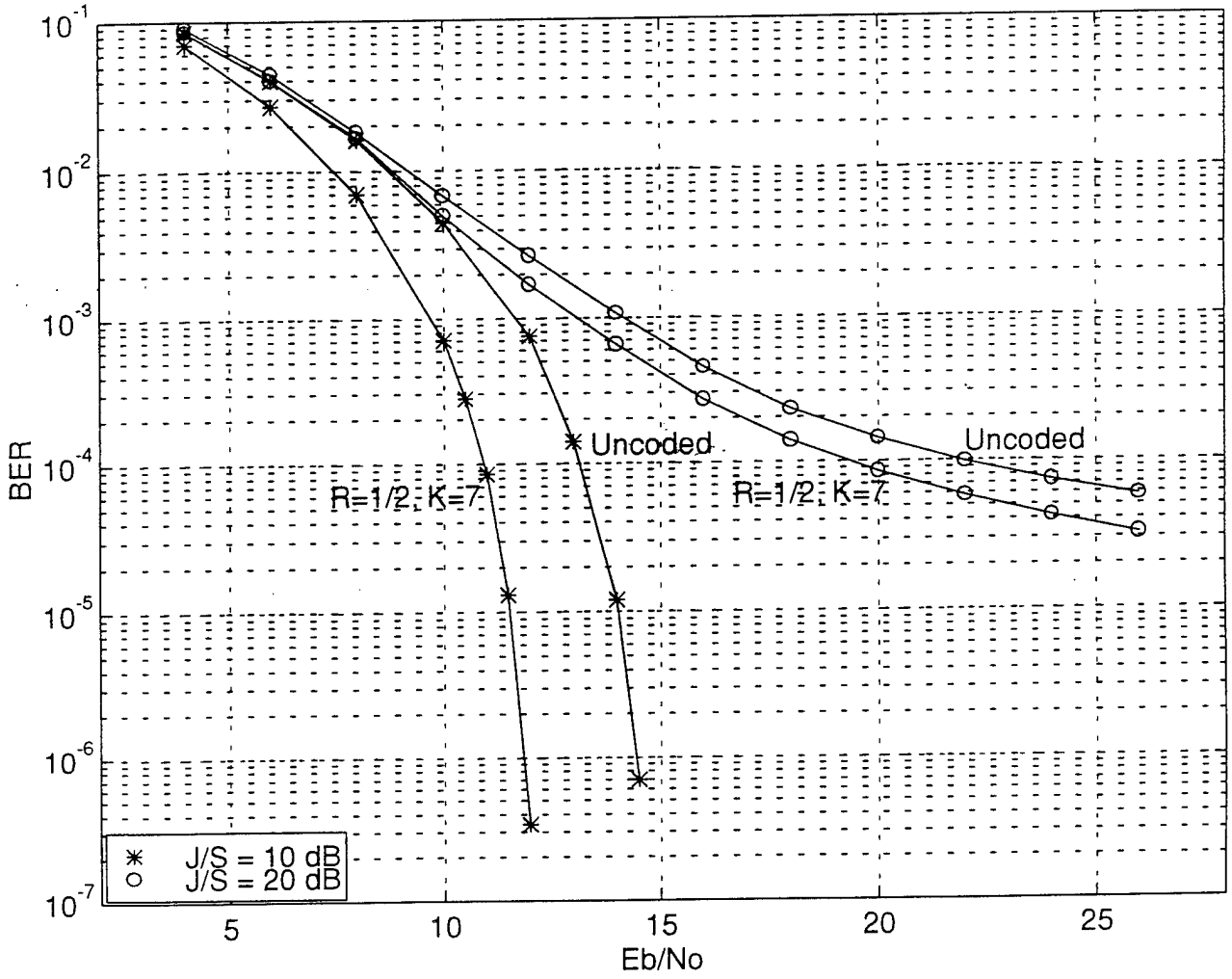

Figure 16. Coded FH/SS-BFSK with UWBS PRF = 40 kHz and T = 5 ns in AWGN Channel

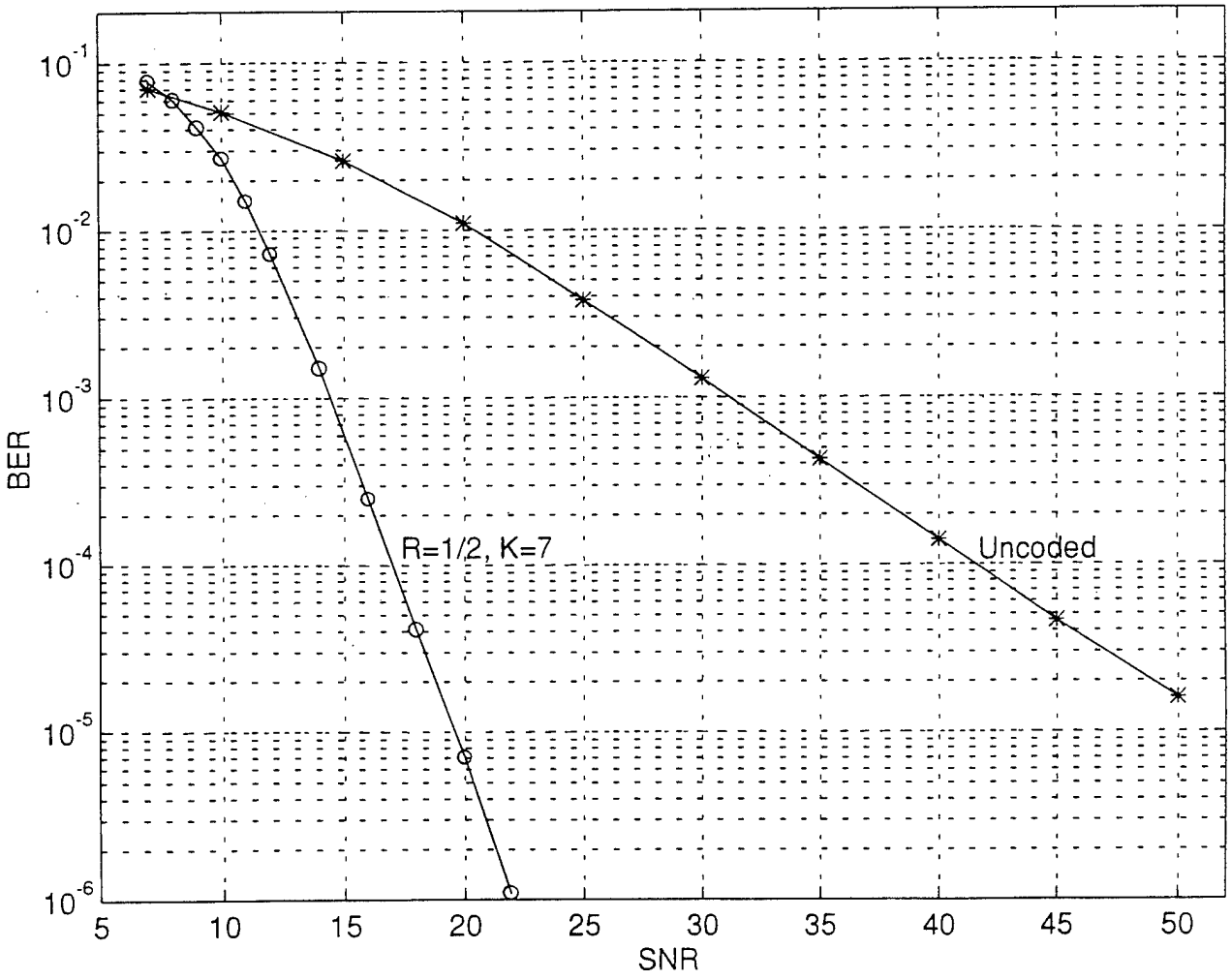

Figure 17. Coded FH/SS-BFSK with UWBS J/S = 6 dB, PRF = 40 kHz and T = 5 ns in Rayleigh Channel

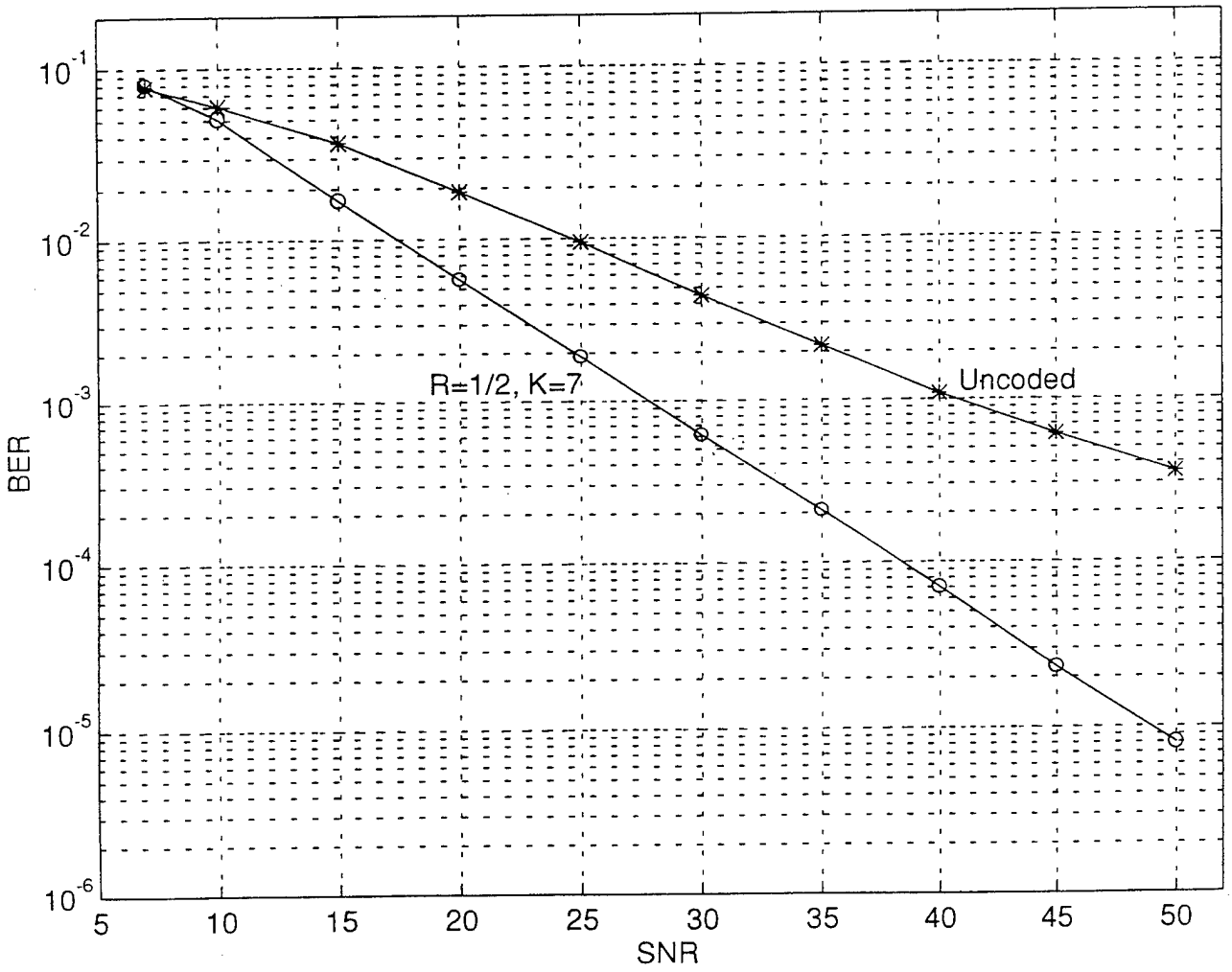

Figure 18. Coded FH/SS-BFSK with UWBS J/S = 16 dB, PRF = 40 kHz and T = 5 ns in Rayleigh Channel

IV. DISCUSSIONS

A. UNCODED DS/SS-BPSK SYSTEM WITH AN UWBS INTERFERENCE

Figure 1 shows the anti-jamming capability of the DS/SS-BPSK system to combat an UWBS interference in the AWGN channel. At J/S = 10 dB, the simulation result was in fairly close agreement with the bit error probability of a conventional coherent BPSK detector as given by [Ref. 6]

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_o}}\right) \tag{4.1}$$

where Q(x) is the Gaussian integral function of x [Ref. 6].

At J/S = 20 dB, the adverse effect of the UWBS interference on the system performance became significant. Compared with the former case where J/S = 10 dB, a degradation of about 17 dB in the SNR was observed at $P_b = 10^{-5}$. It was also noticed that this curve tends to flatten out as the E_b/N_o increases indicating that the jammer has caused some irreducible errors. One possible technique to reduce this error probability is to increase the processing gain of the DS/SS system.

At J/S = 30 dB, the system performance was poor as the bit error probability was greater than 10^{-2} for the whole range of E_b/N_o under consideration. It is evident that the system is overly stressed due to the UWBS interference.

In Figure 2, we examine the jamming effect due to variations in the PRF of the UWBS interference. Three PRF values were considered and it can be seen that as the PRF

increases, the jamming signal inflicts more harm on the communication system. As an illustration, it was found that when the PRF was doubled from 10 kHz to 20 kHz, the SNR degradation was about 1 dB at $P_b = 10^{-5}$. As the PRF was further doubled from 20 kHz to 40 kHz, the SNR degradation became slightly more than 1 dB at the same P_b. It should be noted at this point that the data rate was assumed to be 20 kbps in this report. As such, PRFs of 10 kHz, 20 kHz and 40 kHz would correspond to ½, 1 and 2 UWBS pulses per data bit.

In Figure 3, we look at the effect on the error probabilities due to variations in the UWBS pulse width (T). Three values of T were considered and it was evident that as the UWBS pulses get narrower, the jamming effect becomes more prominent and is thus more detrimental to the system performance. For example at $P_b = 10^{-5}$, it can be seen that as T was reduced from 20 ns to 10 ns, a SNR degradation of about 1 dB was observed. When T was further reduced from 10 ns to 5 ns, about the same SNR degradation of 1 dB was noted. As we know that the null-to-null bandwidth of the UWBS signal is given by

$$B_J = \frac{2}{T} \tag{4.2}$$

an UWBS signal of narrower pulse width would therefore occupy a broader null-to-null bandwidth in the frequency domain.

The preceding paragraphs of this chapter described the system performance for transmission over the AWGN channel. In the next Figures 4 to 6, we will focus our attention on the Rayleigh fading channel which has a time-variant impulse response.

In Figure 4, we are interested to examine the anti-jamming capability of the DS/SS-BPSK system in the Rayleigh fading channel and the three J/S ratios of interest were 6 dB, 16 dB and 26 dB. In the absence of interference, the exact uncoded bit error probability of the DS/SS-BPSK system in the Rayleigh fading channel is given by [Ref. 2]

$$P_b = \frac{1}{2}\left(1 - \sqrt{\frac{SNR}{1 + SNR}}\right) \tag{4.3}$$

where SNR represents the E_b/N_o ratio at the receiver input.

However in the presence of UWBS interference with J/S = 6 dB, it was found that the system performance is slightly worse than that of Equation 4.3. For example at $P_b = 10^{-4}$, the SNR difference between them was about 2 dB.

At J/S = 16 dB, the adverse effect of the UWBS interference on the system performance became prominent. Compared with the former case where J/S = 6 dB, a more serious SNR degradation of 10 dB was observed at $P_b = 10^{-3}$. Similarly as in Figure 1, it can be seen that at higher J/S ratios, the curve tends to flatten out as the E_b/N_o increases indicating irreducible errors caused by the jammer. At J/S = 26 dB, the adverse effect of the UWBS interference was even more profound resulting in an unacceptable system performance of high error rate.

In Figure 5, the jamming effect due to variations in the PRF was examined. Two PRF values were considered and it is shown once again that given a constant jammer power, the jammer can cause more harm by operating at a higher PRF. For example, when the PRF was increased from 10 kHz to 40 kHz, the system performance degraded by about 1.5 dB at $P_b = 10^{-3}$.

On the other hand in Figure 6, the jamming effect due to variations in the UWBS pulse width (T) was examined. Two values of T were considered and the jammer power was assumed to be the same in both cases. At $P_b = 10^{-3}$, a SNR degradation of slightly more than 1 dB was registered when T was reduced from 20 ns to 5 ns. Likewise as in the case of the AWGN channel, it is clear that the communication system is more susceptible to the UWBS interference as its pulses get narrower.

B. CODED DS/SS-BPSK SYSTEM WITH AN UWBS INTERFERENCE

The preceding section has shown that even with the spread spectrum technique, the UWBS jamming could still be fairly effective in degrading the system performance at high J/S ratios. The jamming effect could however be mitigated by employing powerful forward error correction (FEC). In this section, we examine the improvement of a coded system with the convolutional code of rate, R = ½ and a constraint length, K = 7.

Figure 7 presents the performance of coded DS/SS-BPSK system in the AWGN channel with an UWBS interference. Two J/S ratios of 10 dB and 20 dB were considered and we are interested to find out the coding gain achieved with the convolutional code. For J/S = 10 dB, a coding gain of 3 dB was observed at $P_b = 10^{-5}$. For J/S = 20 dB, the coding gain was slightly less than 3 dB at the same P_b.

These simulation results were consistent with the theoretical coding gain achievable by a convolutional code over an uncoded BPSK system. It is given by [Refs. 7 and 8]

$$Coding\ gain \leq 10 \log(R\ d_{free}) \tag{4.4}$$

where d_{free} is the minimum free distance. It should be noted that Equation 4.4 is based on a soft-decision Viterbi decoding. A hard-decision decoding was however used in this thesis and by a general rule of thumb, the coding gains should be reduced by approximately 2 dB in the AWGN channel.

In Figures 8 and 9, we examine the coding gains achieved by the same convolutional code in the Rayleigh fading channel. J/S ratios of 6 dB and 16 dB were considered in these Figures respectively. In Figure 8, significant coding gains of about 13 dB and 29 dB were obtained at P_b of 10^{-3} and 10^{-5} respectively. This result demonstrates that coding is a powerful technique to overcome the SNR degradation due to fading and to combat an UWBS interference. In Figure 9 where J/S ratio was increased to 16 dB, the coding gain achieved was reduced to about 8 dB at $P_b = 10^{-3}$. The convolutional code in this case was still able to provide a good SNR improvement but it was evident that as the jamming effect became more prominent, the coding gain achievable is clipped as it suffered a significant reduction.

C. UNCODED FH/SS-BFSK SYSTEM WITH AN UWBS INTERFERENCE

In the preceding Sections A and B of this chapter, we dealt with the DS/SS-BPSK system in the presence of an UWBS jammer. In the following Sections C and D, we will focus on the other major category of the spread spectrum system, namely the FH/SS-BFSK system with the same UWBS interference. Non-coherent detection and orthogonal signaling techniques were assumed.

Firstly in Figure 10, we will analyze the simulation result for the FH/SS-BFSK system in the AWGN channel with an UWBS interference. At J/S = 10 dB, it was about 1 dB worse than the bit error probability of a conventional non-coherent BFSK detector which is given by

$$P_b = \frac{1}{2} e^{-\frac{E_b}{2N_o}} \tag{4.5}$$

At J/S = 20 dB, we see once again the familiar effect of an UWBS jammer on the system performance as the interference became more prominent. Compared with the former curve where J/S = 10 dB, a degradation of about 9 dB in SNR was observed at $P_b = 10^{-4}$. At J/S = 30 dB, the communication system was crippled as the error rate was unacceptably high.

In Figure 11, we examine the jamming effect due to variations in the PRF of the UWBS interference. Two PRFs were considered and it can be seen that as the PRF increases, the jammer inflicts more harm on the communication system. As shown in the Figure when the PRF was increased from 10 kHz to 40 kHz, the SNR degradation was about 2.5 dB at $P_b = 10^{-4}$. In Figure 12, we look at the effect on the error probabilities due to variations in the UWBS pulse width (T). Two values of T were considered and it was evident that as the UWBS pulses get narrower, the jammer causes more harm to the communication system too as in the previous scenario depicted in Figure 11.

For example at $P_b = 10^{-4}$, it can be seen that as T was reduced from 20 ns to 5 ns, a SNR degradation of about 2 dB was observed. From the jammer's point of view, the results in Figures 11 and 12 provide a better insight as to how to achieve a more effective jamming strategy.

Having examined the simulation results in the AWGN channel, we will turn our attention next on the Rayleigh fading channel in the following Figures 13 to 15. In Figure 13, the anti-jamming capability of the FH/SS-BFSK system in the Rayleigh fading channel was presented where three J/S ratios of 6 dB, 16 dB and 26 dB were considered. At J/S = 6 dB, a bit error probability of 10^{-3} could only be achieved at a relatively high SNR of 31 dB. At J/S = 16 dB, the adverse effect of the UWBS interference on the system performance was even more prominent. As an illustration, to achieve the same P_b of 10^{-3} as in the case where J/S = 6 dB, it requires an increase in SNR of 11 dB i.e., SNR = 42 dB. At J/S = 26 dB, the result shows that effective communication could not be established as the J/S ratio was too high for the spread spectrum system to suppress it effectively.

In Figure 14, the jamming effect due to variations in the PRF of the UWBS interference was examined. Two PRF values were considered and the result shows that the jammer can do more harm when operating at a higher PRF given a constant jammer power. As an illustration, when the PRF was increased from 10 kHz to 40 kHz, the system performance degraded by about 3 dB at $P_b = 10^{-3}$.

In Figure 15, the jamming effect due to variations in the UWBS pulse width (T) was examined. Two values of T were considered and at $P_b = 10^{-3}$, a SNR degradation of about 2.5 dB was observed when T was reduced from 20 ns to 5 ns.

D. CODED FH/SS-BFSK SYSTEM WITH AN UWBS INTERFERENCE

We have seen in Section B of this chapter how the UWBS jamming effect on the DS/SS-BPSK system could be mitigated by employing the convolutional code of rate R = ½ and a constraint length K = 7. In this section, we present the coded improvement achieved on the FH/SS-BFSK system using the same convolutional code in the presence of an UWBS jammer.

Figure 16 shows the coded performance of the FH/SS-BFSK system in the AWGN channel. Two J/S ratios of 10 dB and 20 dB were considered and it was found that coding gains of 2.5 dB (at $P_b = 10^{-5}$) and 1 dB (at $P_b = 10^{-3}$) were achieved at J/S = 10 dB and 20 dB respectively.

In Figures 17 and 18, we examine the coding gains achieved by the same convolutional code in the Rayleigh fading channel. J/S ratios of 6 dB and 16 dB were considered in these Figures respectively. As in case of the coded DS/SS-BPSK system in the Rayleigh fading channel described in Section B of this chapter, the use of coding has once again proved to be a very effective technique to overcome the SNR degradation due to fading and to combat an UWBS interference.

As evident in Figure 17, a significant coding gain of 24 dB was achieved at $P_b = 10^{-4}$. Lastly in Figure 18, J/S ratio was increased to 16 dB indicating a stronger received jamming power. The convolutional code was still able to achieve a fairly good though lower coding gain of 12 dB at $P_b = 10^{-3}$.

E. COMPARISON OF UWBS JAMMING EFFECTS ON DS/SS-BPSK AND FH/SS-BFSK SYSTEMS

Sections A and B of this chapter described the UWBS jamming effect on the DS/SS-BPSK system whereas Sections C and D discussed the UWBS jamming effect on the FH/SS-BFSK system. In this section, we would like to examine whether the UWBS jammer is more effective on the DS/SS-BPSK or the FH/SS-BFSK systems given the same jammer power and parameters.

In comparing Figures 1 and 10 where the PRF and T were fixed at 40 kHz and 5 ns respectively, it was observed that when the J/S ratio was increased from 10 dB to 20 dB, the jammer caused SNR degradations of 4.5 dB and 9 dB in the DS/SS-BPSK and the FH/SS-BFSK systems respectively at $P_b = 10^{-4}$. This result indicates that the FH/SS-BFSK system has a higher degree of susceptibility to an increase in the J/S ratio.

The above comparison was based on the AWGN channel. We can make a similar comparison for the Rayleigh fading channel by examining Figures 4 and 13 where the PRF and T were again fixed at 40 kHz and 5 ns respectively. It was found that when the J/S ratio was increased from 6 dB to 16 dB, the SNR degradations were 10 dB and 11 dB in the DS/SS-BPSK and the FH/SS-BFSK systems respectively at $P_b = 10^{-3}$.

This result is consistent with that found earlier for the AWGN channel which indicates that the DS/SS-BPSK system has higher tolerance against an UWBS interference compared with the FH/SS-BFSK system.

V. CONCLUSION

The concept of jamming of the spread spectrum communication systems and their performances in different jamming environment are well documented [Refs. 1 - 5]. Barrage noise jamming, pulsed noise jamming, partial band jamming and tone jamming are some of the common jamming strategies that can be adopted to reduce and prevent the use of certain portion of the electromagnetic spectrum by other parties. However, the potential jamming capability of an UWBS jammer was not widely studied in the literature because the development and availability of the UWBS device were fairly recent. The main motivation of this research was thus to investigate the impact of UWBS jamming on spread spectrum systems.

Simulation results of the error probabilities for the DS/SS-BPSK and the FH/SS-BFSK systems in an UWBS interference were presented and discussed in Chapters III and IV. The AWGN channel and the Rayleigh fading channel were considered with and without convolutional code. The findings and results have enabled us to find answers to the key research questions and thus achieved the goals of this thesis. We have shown that given the same jammer power, an UWBS interference is more effective on the FH/SS-BFSK system than the DS/SS-BPSK system in causing a higher bit error probability. From the jammer's point of view, this report provides an insight as to how the jammer parameters can be varied in order to inflict more harm on the communication systems. In particular, we have shown that given the same jammer power, the jammer is more effective by operating at a higher PRF or with a narrower pulse width. In addition, the coding gains

achieved with the convolutional code to reduce the UWBS jamming effect in both the AWGN channel and the Rayleigh fading channel were presented. The results have shown that the coding gain achieved in the Rayleigh fading channel is at least an order higher than that achieved in the AWGN channel at $P_b = 10^{-4}$ or better.

This thesis concludes the pioneer effort to analyze the UWBS jamming capability on spread spectrum systems using the ACOLADE simulation tool. However, due to its infancy, several critical software bugs were encountered during the course of the research work. Nevertheless, they were systematically debugged to ensure smooth executions of the simulation programs. As these hitches have been successfully ironed out, the scope of future work using the ACOLADE tool to investigate the UWBS jamming effect on communication systems can easily be expanded, e.g., to include other types of fading channels or other FEC codes.

LIST OF REFERENCES

1. R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread Spectrum Communications - A Tutorial," *IEEE Trans. on Communications*, Vol. COM-30, No. 5, pp. 855-884, May 1982.

2. M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, *Spread Spectrum Communications Handbook*, McGraw-Hill, Inc., Los Angeles, CA, 1994.

3. R. L. Peterson, R. E. Ziemer, and D. E. Borth, *Introduction to Spread Spectrum Communications*, Prentice Hall, Inc., Atlanta, GA, 1995.

4. R. C. Dixon, *Spread Spectrum Systems*, John Wiley & Sons, Inc., Newark, NJ, 1984.

5. D. L. Schilling, L. B. Milstein, R. L. Pickholtz, and R. W. Brown, "Optimization of the Processing Gain of an M-ary Direct Sequence Spread Spectrum Communication System," *IEEE Trans. on Communications*, Vol. COM-28, No. 8, August 1980.

6. J. G. Proakis, *Digital Communications*, McGraw-Hill, Inc., Los Angeles, CA, 1995.

7. S. Lin and D. J. Costello, Jr., *Error Control Coding : Fundamentals and Applications*, Prentice Hall, Inc., Atlanta, GA, 1983.

8. G. C. Clark, Jr. and J. B. Cain, *Error Correction Coding for Digital Communications*, Plenum Press, New York, NY, 1981.

9. B. Edde, *Radar : Principles, Technology and Applications*, Prentice Hall, Inc., Atlanta, GA, 1993.

10. *ACOLADE User's Manual*, ICUCOM Corporation, 1993.

11. *ACOLADE Tutorial Manual*, ICUCOM Corporation, 1994.

APPENDIX A. PROGRAM LISTING OF "DIRECT SEQUENCE SPREAD SPECTRUM" MODEL IN THE C LANGUAGE

The ACOLADE simulation is based on the concept of models. As an example, the "Direct Sequence Spread Spectrum" model used in this thesis, written in the C language, is listed below.

```
*
* The objective of this program is to implement the direct sequence spread spectrum model.
* A random chipping  sequence on the I and Q channels are created as the pseudo random
* waveform to achieve the desired spectrum spreading.
*
*/
#include <stdio.h>
#include <string.h>
#include <math.h>
#include "acolade.h"

typedef struct
{
   INTFOUR table[50];
   INTFOUR ns_chip;
   INTFOUR block;
   int chip_count;
   ACO_REAL_DATA i_chip;
   ACO_REAL_DATA q_chip;
} SV, *SV_PTR;

ACO_MODULE ssds_d(POINTER modhandle)
{
   /* Declare the called subroutines. */
   void_f for_decl rng1rqq();
   void_f for_decl rngin2qq();

   /* Declare the local variables. */
   INTFOUR nbytes;
   INTFOUR i1, i2, i3, i4;
   int I;
   INTFOUR portnum = 0;
   int sim_state;
```

```
    float real_random;
    ACO_REAL_DATA *in, *out;
    SV_PTR SP;

    sim_state = (int) simstate();

if(sim_state == SIM_INIT)
{
    nbytes = (INTFOUR) sizeof(SV);
    SP = (SV_PTR) setstvar(modhandle, &nbytes);

    /* Read the number of samples per chip. */
    get_prm(modhandle, 0, INTEGER, 1, &SP->ns_chip);

    /* Read in the blocking factor. */
    get_prm(modhandle, 1, INTEGER, 1, &SP->block);

    /* Multiply the blocking factor by the number of samples per chip
    to determine the total number of samples to be processed. */
    SP->block *= SP->ns_chip;

    /* Set up the data stacks:
    Input. */
    istblksiz(modhandle, &portnum, &SP->block);

    /* Output. */
    ostblksiz(modhandle, &portnum, &SP->block);

    return(0);
}

else if(sim_state == PARAM_SET)
{
    SP = (SV_PTR) setstvar(modhandle,&nbytes);

    /* Fetch the input seeds. */
    get_prm(modhandle, 2, INTEGER, 1, &i1);
    get_prm(modhandle, 3, INTEGER, 1, &i2);
    get_prm(modhandle, 4, INTEGER, 1, &i3);
    get_prm(modhandle, 5, INTEGER, 1, &i4);
    rngin2qq(SP->table,&i1,&i2,&i3,&i4);

    /* (Re)set the chip count to zero. */
    SP->chip_count = 0;

}
else if(sim_state == SIMULATE)
```

```
{
   if(((int) stacksin(modhandle)) > 0)
   {
      /* Reacquire state variable addresses. */
      SP = (SV_PTR) setstvar(modhandle, &nbytes);

      /* Reacquire buffer addresses. */
      get_buf_in(modhandle, &portnum, &in);
      get_buf_out(modhandle, &portnum, &out);

      /* Loop over the number of input samples. */
      for(I=0; i<SP->block; I++)
      {
         /* Get new random bits if the current sample is the
         first of a chip interval. */
         if(SP->chip_count == (int) 0)
         {
            /* Set the real chip. */
            rng1rqq(SP->table,&real_random);
            if(real_random > 0.5)
               SP->i_chip = 1.0;
            else
               SP->i_chip = -1.0;

            /* Now the imaginary chip. */
            rng1rqq(SP->table,&real_random);
            if(real_random > 0.5)
               SP->q_chip = 1.0;
            else
               SP->q_chip = -1.0;
         }

         /* Increment the chip count for next time. */
         SP->chip_count = (++SP->chip_count)%(SP->ns_chip);

         /* Multiply the (complex) input sample by the I
         and Q chips to determine the output. */
         out[2*I] = SP->i_chip * in[2*I];
         out[2*I+1] = SP->q_chip * in[2*I+1];
      }

      /* Output the stacks. */
      stacksout(modhandle);
   }
}
return(0);
}
```

APPENDIX B. LISTING OF A NET FILE NAMED DS_AWGN.NET

This Appendix shows the listing of a NET file named DS_AWGN.net. It was created to be used in the ACOLADE simulation environment in order to obtain the error probabilities of the DS/SS-BPSK system with an UWBS interference in the AWGN channel. The corresponding simulation results have been presented earlier in this report in Figure 1.

```
* START OF DS_AWGN.NET FILE
*
 .ICON,ICON_SIZE={0,0},ICON_ROTATION=0,SCREEN_STATE={0,0,554,0},
 IPIN_SPACE=SPACED,OPIN_SPACE=SPACED,EXT_COLOR=BLACK,INT_COLOR=BLACK,
  TEXT_COLOR=BLACK,PIN_EXT_COLOR=BLACK,PIN_INT_COLOR=BLACK
 .PARAMETER,TYPE=INTEGER,VALUE=(*>0)32,
 PARNAME="Basic System Sampling Rate",PARSYM="@samplerate_0@"
 .PARAMETER,TYPE=INTEGER,VALUE=(*>0)1000,PARNAME="Blocking Factor",
 PARSYM="@block@"
 .PARAMETER,TYPE=REAL,VALUE=0.0,
 PARNAME="System SNR per Channel Symbol (dB)",PARSYM="@Es/N0(dB)@"
 .ENDGPARAM
.MODULE,MODULE_NAME="control",INSTANCE_NAME="control",
CLASS_NAME="control",FILE_NAME="cntl\cntl_d",FUNC_NAME="cntl_d",
 NINPUTS=0,NOUTPUTS=0,NPARAMS=3,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,
  PRECISION=0,ROUND_CHAR=0,OVF_CHAR=0
 .ICON,ICON_SIZE={128,32},ICON_POSITION={510,832},ICON_ROTATION=0,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,OPIN_SPACE=SPACED,
  EXT_COLOR=DARK_GREY,INT_COLOR=CYAN,BLOCK_NAME="Executive%Control",
   TEXT_COLOR=BLUE,PIN_SIZE=9,PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,
    ICON_FILE="iconfile",ICON_INDEX=1
 .PARAMETER,TYPE=CHAR_STRING,VALUE=["Es/N0(dB)"]"Es/N0(dB)",
 HELPTEXT="Global Parameter Facility symbol name.",
  PARNAME="GPF_Symbol_Name"
 .PARAMETER,TYPE=REAL,VALUE=[0.0]3.0,
 HELPTEXT="The initial value for the iterated parameter.",
  PARNAME="Initial_Parameter_Value"
 .PARAMETER,TYPE=REAL,VALUE=[1.0]1.0,
 HELPTEXT="The increment for the iterated parameter.",
  PARNAME="Parameter_Increment"
 .ENDNODE
.MODULE,MODULE_NAME="dsrc",INSTANCE_NAME="dsrc",CLASS_NAME="SOURCES",
FILE_NAME="dsrc\dsrc_d",FUNC_NAME="dsrc_d",NINPUTS=0,NOUTPUTS=1,
 NPARAMS=5,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,
  OVF_CHAR=0
 .ICON,ICON_SIZE={64,32},ICON_POSITION={112,48},ICON_ROTATION=0,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,OPIN_SPACE=SPACED,
```

```
 OPIN_COOR={16,0},EXT_COLOR=BLUE,INT_COLOR=GREEN,
  BLOCK_NAME="Digital%Source",TEXT_COLOR=BLUE,PIN_SIZE=9,
   PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,ICON_FILE="iconfile",
    ICON_INDEX=1
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
 HELPTEXT="Initial state for integer random number generator.",
  PARNAME="Seed"
.PARAMETER,TYPE=INTEGER,VALUE=[1]1,
 HELPTEXT="Logarithm (base 2) of alphabet size",PARNAME="Log2_of_M",
  PARSYM="@log2_of_alphabet_size@"
.PARAMETER,TYPE=INTEGER,VALUE=[-1]-1,
 HELPTEXT="Maximum number of symbols to generate",PARNAME="Max_Syms"
.PARAMETER,TYPE=INTEGER,VALUE=[1]1,
 HELPTEXT="1: Stop simulation when Max_Syms reached.",
  PARNAME="Termination_Flag"
.PARAMETER,TYPE=INTEGER,VALUE=[64]@block@,
 HELPTEXT="Number of symbols generated per pass.",PARNAME="Num_Symbols"
.OUTPUT,PORT_NUM=0,DATA_SIZE=4,DATA_TYPE=DIGITAL_DATA,
 SAMPLE_RATE=-(2^(@log2_of_alphabet_size@)),WAVE_FILE_NAME="*.src"
.ENDNODE
.SUBNET,SUBNET_NAME="xbpsk",INSTANCE_NAME="xbpsk",CLASS_NAME="xmtrs",
 FILE_NAME="SUBNET_DIRS:xbpsk",FUNC_NAME=" ",NINPUTS=1,NOUTPUTS=1,
  NPARAMS=17,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,
   OVF_CHAR=0
.ICON,ICON_SIZE={67,34},ICON_POSITION={225,48},ICON_ROTATION=0,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,IPIN_COOR={0,0},
  OPIN_SPACE=SPACED,OPIN_COOR={0,0},EXT_COLOR=BLUE,INT_COLOR=GREEN,
   BLOCK_NAME="BPSK%Transmitter",TEXT_COLOR=BLUE,PIN_SIZE=9,
    PIN_EXT_COLOR=BLACK,PIN_INT_COLOR=YELLOW,ICON_FILE="acolade.icn",
     ICON_INDEX=2008
.PARAMETER,TYPE=INTEGER,VALUE=(0.0<*)[@block@]@block@,
 HELPTEXT="Number of symbols to process per invocation",
  PARNAME="Num_Symbols"
.PARAMETER,TYPE=INTEGER,VALUE=(*>0)[8]@samplerate_0@,
 HELPTEXT="Number of Samples per Channel Signalling Interval",
  PARNAME="Sample_Rate"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<=6)[0]0,
 HELPTEXT="0=NONE,1=FILE,2=RCOS,3=RTRCOS,4=GMSK,5=HALFSIN,6=GAUSS",
  PARNAME="I_Pulse_Selector"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<=6)[0]0,
 HELPTEXT="0=NONE,1=FILE,2=RCOS,3=RTRCOS,4=GMSK,5=HALFSIN,6=GAUSS",
  PARNAME="Q_Pulse_Selector"
.PARAMETER,TYPE=REAL,VALUE=[{1.0,1.0}]{1.0,1.0},
 HELPTEXT="Pulse shape characteristics array",
  PARNAME="I_Pulse_Parameters"
.PARAMETER,TYPE=REAL,VALUE=[{1.0,1.0}]{1.0,1.0},
 HELPTEXT="Pulse shape characteristics array",
  PARNAME="Q_Pulse_Parameters"
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
 HELPTEXT="Multiple of preceding symbols that the modulating pulse overlaps.",
  PARNAME="I_Future_Pulse_Skirt_Length"
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
 HELPTEXT="Multiple of preceding symbols that the modulating pulse overlaps.",
  PARNAME="Q_Future_Pulse_Skirt_Length"
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
 HELPTEXT="Multiple of succeding symbols that the modulating pulse overlaps.",
  PARNAME="I_Past_Pulse_Skirt_Length"
```

```
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
HELPTEXT="Multiple of succeding symbols that the modulating pulse overlaps.",
 PARNAME="Q_Past_Pulse_Skirt_Length"
.PARAMETER,TYPE=DATA_FILE,VALUE=["PULS_DIR:pulse.pul"]"*",
HELPTEXT="File of complex pulse sample values. ",
 PARNAME="I_Pulse_Filename"
.PARAMETER,TYPE=DATA_FILE,VALUE=["PULS_DIR:pulse.pul"]"*",
HELPTEXT="File of complex pulse sample values. ",
 PARNAME="Q_Pulse_Filename"
.PARAMETER,TYPE=REAL,VALUE=0.0,
HELPTEXT="Time offset between I and channels in multiples of a symbol.",
 PARNAME="Q_Channel_Delay"
.PARAMETER,TYPE=REAL,VALUE=0.0,
HELPTEXT="Phase offset between I and channels in radians.",
 PARNAME="Q_Phase_Offset"
.PARAMETER,TYPE=REAL,VALUE=[@Es/N0(dB)@]@Es/N0(dB)@,
HELPTEXT="Transmitted signal energy per signalling interval (in dB).",
 PARNAME="Signal_Level_in_dB"
.PARAMETER,TYPE=REAL,VALUE=[0.0]0.0,
HELPTEXT="Used forever if phase port unconnected.",
 PARNAME="Initial_Phase_Offset"
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
HELPTEXT="Used only for random phase option.",
 PARNAME="Initial_Random_Seed"
.INPUT,PORT_NUM=0,DATA_SIZE=4,DATA_TYPE=DIGITAL_DATA,SAMPLE_RATE=-2.0
.OUTPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,SAMPLE_RATE=0.0
.ENDNODE
.MODULE,MODULE_NAME="ssds",INSTANCE_NAME="ssds",CLASS_NAME="SPRD_SPC",
FILE_NAME="ssds\ssds_d",FUNC_NAME="ssds_d",NINPUTS=1,NOUTPUTS=1,
 NPARAMS=6,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,
  OVF_CHAR=0
.ICON,ICON_SIZE={63,32},ICON_POSITION={351,48},ICON_ROTATION=0,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,IPIN_COOR={-16,0},
  OPIN_SPACE=SPACED,OPIN_COOR={16,0},EXT_COLOR=BLUE,INT_COLOR=GREEN,
   BLOCK_NAME="Direct%Sequence%Sprd. Spec.",TEXT_COLOR=BLUE,PIN_SIZE=9,
    PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,ICON_FILE="iconfile",
     ICON_INDEX=1
.PARAMETER,TYPE=INTEGER,VALUE=[2]2,
 HELPTEXT="Number of Samples per Chip",PARNAME="Chip_Sample_Rate"
.PARAMETER,TYPE=INTEGER,VALUE=[@block@]1450,
 HELPTEXT="Number of chips processed per invocation.",
  PARNAME="Num_Chips"
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_1"
.PARAMETER,TYPE=INTEGER,VALUE=0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_2"
.PARAMETER,TYPE=INTEGER,VALUE=0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_3"
.PARAMETER,TYPE=INTEGER,VALUE=0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_4"
.INPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA
.OUTPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,
 SAMPLE_RATE=@samplerate_0@,WAVE_FILE_NAME="*.spd"
.ENDNODE
.MODULE,MODULE_NAME="cwgn",INSTANCE_NAME="cwgn",CLASS_NAME="sources",
FILE_NAME="wgn1\wgn1_d",FUNC_NAME="wgn1_d",NINPUTS=0,NOUTPUTS=1,
 NPARAMS=8,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,
```

```
 OVF_CHAR=0
.ICON,ICON_SIZE={64,32},ICON_POSITION={648,85},ICON_ROTATION=180,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,OPIN_SPACE=SPACED,
  OPIN_COOR={16,0},EXT_COLOR=BLUE,INT_COLOR=GREEN,
   BLOCK_NAME="Complex White%Gauss. Noise%Source",TEXT_COLOR=BLUE,
    PIN_SIZE=9,PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,
     ICON_FILE="iconfile",ICON_INDEX=1
.PARAMETER,TYPE=REAL,VALUE=[0.0]0.0,
 HELPTEXT="dB value of power density flat on (-Ns/2, Ns/2).",
  PARNAME="Power_Spectral_Density_in_dB"
.PARAMETER,TYPE=INTEGER,VALUE=[@samplerate_0@]@samplerate_0@,
 HELPTEXT="Number of samples per symbol.",PARNAME="Sample_Rate"
.PARAMETER,TYPE=INTEGER,VALUE=[@block@]@block@,
 HELPTEXT="Number of symbols to generate during each invocation.",
  PARNAME="Num_Symbols"
.PARAMETER,TYPE=INTEGER,VALUE=[-1]-1,HELPTEXT="-1: Run forever.",
 PARNAME="Max_Blocks_to_Generate"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",PARNAME="Seed_1"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",PARNAME="Seed_2"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",PARNAME="Seed_3"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",PARNAME="Seed_4"
.OUTPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA
.ENDNODE
.MODULE,MODULE_NAME="unlu",INSTANCE_NAME="unlu",CLASS_NAME="sources",
FILE_NAME="unip\unlu_d",FUNC_NAME="unlu_d",NINPUTS=0,NOUTPUTS=1,
 NPARAMS=17,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,
  OVF_CHAR=0
.ICON,ICON_SIZE={64,32},ICON_POSITION={457,92},ICON_ROTATION=0,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,OPIN_SPACE=SPACED,
  OPIN_COOR={16,0},EXT_COLOR=BLUE,INT_COLOR=GREEN,
   BLOCK_NAME="Log-Uniform%Square Pulse%Source",TEXT_COLOR=BLUE,
    PIN_SIZE=9,PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,
     ICON_FILE="iconfile",ICON_INDEX=1
.PARAMETER,TYPE=INTEGER,VALUE=(0<*)[@samplerate_0@]@samplerate_0@,
 HELPTEXT="Number of samples per symbol.",PARNAME="Sample_Rate"
.PARAMETER,TYPE=INTEGER,VALUE=(0<*)[@block@]@block@,
 HELPTEXT="Number of symbols to generate during each invocation.",
  PARNAME="Num_Symbols"
.PARAMETER,TYPE=INTEGER,VALUE=[-1]-1,HELPTEXT="-1: Run forever.",
 PARNAME="Max_Blocks_to_Generate"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<=1)[0]0,
 HELPTEXT="1: Random impulse phase, 0: Zero impulse phase",
  PARNAME="Random_Phase_Flag"
.PARAMETER,TYPE=REAL,VALUE=(0.0<*)[1.0]0.0002,
 HELPTEXT="Pulse_Duration_in_Symbols.",PARNAME="Pulse_Duration"
.PARAMETER,TYPE=REAL,VALUE=(0.0<*)[0.5]1,
 HELPTEXT="Average number of hits per symbol.",
  PARNAME="Impulse_Arrival_Rate"
.PARAMETER,TYPE=INTEGER,VALUE=[0]-1,
 HELPTEXT="-1: Periodic arrivals, 0: Random arrivals",
  PARNAME="Nu_for_Gamma_Density"
.PARAMETER,TYPE=REAL,VALUE=[0.0]10.0,
 HELPTEXT="Lower limit for log-uniform distribution.",
```

```
 PARNAME="Lower_Limit_in_dB"
.PARAMETER,TYPE=REAL,VALUE=(0.0<*)[1.0]10.0,
 HELPTEXT="Upper limit for log-uniform distribution.",
 PARNAME="Upper_Limit_in_dB"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_1_for_Arrivals"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_2_for_Arrivals"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_3_for_Arrivals"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_4_for_Arrivals"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_1_for_Phases"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_2_for_Phases"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_3_for_Phases"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Initial state for random number generator.",
 PARNAME="Seed_4_for_Phases"
.OUTPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,
 SAMPLE_RATE=@samplerate_0@,WAVE_FILE_NAME="*.unp"
.ENDNODE
.MODULE,MODULE_NAME="add",INSTANCE_NAME="add",CLASS_NAME="primtive",
FILE_NAME="add\add_d",FUNC_NAME="add_d",NINPUTS=3,NOUTPUTS=1,NPARAMS=1,
 MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,OVF_CHAR=0
.ICON,ICON_SIZE={32,64},ICON_POSITION={556,150},ICON_ROTATION=270,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,
  IPIN_COOR={-16,0,-16,10,-16,-10},OPIN_SPACE=SPACED,OPIN_COOR={16,10},
   EXT_COLOR=BLUE,INT_COLOR=GREEN,BLOCK_NAME="Adder",TEXT_COLOR=BLUE,
    PIN_SIZE=7,PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,
     ICON_FILE="iconfile",ICON_INDEX=1
.PARAMETER,TYPE=INTEGER,VALUE=(0=<*)[0]0,
 HELPTEXT="Number of additions per invocation",PARNAME="Block_Size"
.INPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,SAMPLE_RATE=0.0
.INPUT,PORT_NUM=1,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,SAMPLE_RATE=0.0
.INPUT,PORT_NUM=2,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,SAMPLE_RATE=0.0
.OUTPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,
 SAMPLE_RATE=@samplerate_0@,WAVE_FILE_NAME="*.sum"
.ENDNODE
.MODULE,MODULE_NAME="ssds",INSTANCE_NAME="ssds_1",
CLASS_NAME="SPRD_SPC",FILE_NAME="ssds\ssds_d",FUNC_NAME="ssds_d",
 NINPUTS=1,NOUTPUTS=1,NPARAMS=6,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,
  PRECISION=0,ROUND_CHAR=0,OVF_CHAR=0
.ICON,ICON_SIZE={64,32},ICON_POSITION={351,217},ICON_ROTATION=180,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,IPIN_COOR={-16,0},
  OPIN_SPACE=SPACED,OPIN_COOR={16,0},EXT_COLOR=BLUE,INT_COLOR=GREEN,
   BLOCK_NAME="Direct%Sequence%Sprd. Spec.",TEXT_COLOR=BLUE,PIN_SIZE=9,
    PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,ICON_FILE="iconfile",
```

```
   ICON_INDEX=1
.PARAMETER,TYPE=INTEGER,VALUE=[2]2,
 HELPTEXT="Number of Samples per Chip",PARNAME="Chip_Sample_Rate"
.PARAMETER,TYPE=INTEGER,VALUE=[@block@]1450,
 HELPTEXT="Number of chips processed per invocation.",
  PARNAME="Num_Chips"
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_1"
.PARAMETER,TYPE=INTEGER,VALUE=0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_2"
.PARAMETER,TYPE=INTEGER,VALUE=0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_3"
.PARAMETER,TYPE=INTEGER,VALUE=0,
 HELPTEXT="Seed for random number generator.",PARNAME="Seed_4"
.INPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA
.OUTPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,
 SAMPLE_RATE=@samplerate_0@,WAVE_FILE_NAME="*.spd"
.ENDNODE
.SUBNET,SUBNET_NAME="rbpsk",INSTANCE_NAME="rbpsk",CLASS_NAME="rcvrs",
FILE_NAME="SUBNET_DIRS:rbpsk",FUNC_NAME=" ",NINPUTS=1,NOUTPUTS=3,
 NPARAMS=17,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,
  OVF_CHAR=0
.ICON,ICON_SIZE={74,39},ICON_POSITION={233,215},ICON_ROTATION=180,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,IPIN_COOR={0,0},
  OPIN_SPACE=SPACED,OPIN_COOR={0,0,0,0,0,0},EXT_COLOR=BLUE,
   INT_COLOR=GREEN,BLOCK_NAME="BPSK%Receiver",TEXT_COLOR=BLUE,PIN_SIZE=9,
    PIN_EXT_COLOR=BLACK,PIN_INT_COLOR=YELLOW,ICON_FILE="acolade.icn",
     ICON_INDEX=2008
.PARAMETER,TYPE=INTEGER,VALUE=(0.0<*)[@block@]@block@,
 HELPTEXT="Number of symbols to process per invocation.",
  PARNAME="Num_Symbols"
.PARAMETER,TYPE=INTEGER,VALUE=@samplerate_0@,
 HELPTEXT="Number of samples per symbol.",PARNAME="Sample_Rate"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<=6)[0]0,
 HELPTEXT="0=NONE,1=FILE,2=RCOS,3=RTRCOS,4=GMSK,5=HALFSIN,6=GAUSS",
  PARNAME="I_Pulse_Selector"
.PARAMETER,TYPE=INTEGER,VALUE=(0=<=6)[0]0,
 HELPTEXT="0=NONE,1=FILE,2=RCOS,3=RTRCOS,4=GMSK,5=HALFSIN,6=GAUSS",
  PARNAME="Q_Pulse_Selector"
.PARAMETER,TYPE=REAL,VALUE=[{1.0,1.0}]{1.0,1.0},
 HELPTEXT="Pulse shape characteristics array",
  PARNAME="I_Pulse_Parameters"
.PARAMETER,TYPE=REAL,VALUE=[{1.0,1.0}]{1.0,1.0},
 HELPTEXT="Pulse shape characteristics array",
  PARNAME="Q_Pulse_Parameters"
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
 HELPTEXT="Multiple of preceding symbols that the modulating pulse overlaps.",
  PARNAME="I_Future_Pulse_Skirt_Length"
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
 HELPTEXT="Multiple of preceding symbols that the modulating pulse overlaps.",
  PARNAME="Q_Future_Pulse_Skirt_Length"
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
 HELPTEXT="Multiple of succeding symbols that the modulating pulse overlaps.",
  PARNAME="I_Past_Pulse_Skirt_Length"
.PARAMETER,TYPE=REAL,VALUE=(0.0=<*)[0.0]0.0,
 HELPTEXT="Multiple of succeding symbols that the modulating pulse overlaps.",
  PARNAME="Q_Past_Pulse_Skirt_Length"
```

```
.PARAMETER,TYPE=DATA_FILE,VALUE=["PULS_DIR:pulse.pul"]"*",
 HELPTEXT="File of complex pulse sample values. ",
  PARNAME="I_Pulse_Filename"
.PARAMETER,TYPE=DATA_FILE,VALUE=["PULS_DIR:pulse.pul"]"*",
 HELPTEXT="File of complex pulse sample values. ",
  PARNAME="Q_Pulse_Filename"
.PARAMETER,TYPE=REAL,VALUE=0.0,
 HELPTEXT="Time offset between I and Q channels in multiples of a symbol.",
  PARNAME="Q_Channel_Delay"
.PARAMETER,TYPE=REAL,VALUE=0.0,
 HELPTEXT="I/Q channel phase offset minus PI/2.",
  PARNAME="Q_Phase_Offset"
.PARAMETER,TYPE=REAL,VALUE=[@Es/N0(dB)@]@Es/N0(dB)@,
 HELPTEXT="Received average power.",PARNAME="AGC_Level_in_dB"
.PARAMETER,TYPE=REAL,VALUE=[0.0]0.0,
 HELPTEXT="Initial estimate in radians will be used if no tracking.",
  PARNAME="Initial_Phase_Estimate"
.PARAMETER,TYPE=REAL,VALUE=[0.0]0.0,
 PARNAME="Initial_Bit_Timing_Offset"
.INPUT,PORT_NUM=0,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,SAMPLE_RATE=0.0
.OUTPUT,PORT_NUM=0,DATA_SIZE=4,DATA_TYPE=DIGITAL_DATA,SAMPLE_RATE=0.0
.OUTPUT,PORT_NUM=1,DATA_SIZE=8,DATA_TYPE=COMPLEX_DATA,SAMPLE_RATE=0.0
.OUTPUT,PORT_NUM=2,DATA_SIZE=4,DATA_TYPE=REAL_DATA
.ENDNODE
.MODULE,MODULE_NAME="sect",INSTANCE_NAME="sect",CLASS_NAME="COUNTERS",
FILE_NAME="sect\sect_d",FUNC_NAME="sect_d",NINPUTS=1,NOUTPUTS=1,
 NPARAMS=10,MAXACC=0,FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,
  OVF_CHAR=0
.ICON,ICON_SIZE={64,32},ICON_POSITION={126,214},ICON_ROTATION=180,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,IPIN_COOR={-16,0},
  OPIN_SPACE=SPACED,OPIN_COOR={16,0},EXT_COLOR=BLUE,INT_COLOR=GREEN,
   BLOCK_NAME="Regenerative%Error Counter",TEXT_COLOR=BLUE,PIN_SIZE=9,
    PIN_EXT_COLOR=MAGENTA,PIN_INT_COLOR=YELLOW,ICON_FILE="iconfile",
     ICON_INDEX=1
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
 HELPTEXT="Initial random number generator state.",PARNAME="Seed"
.PARAMETER,TYPE=INTEGER,VALUE=[1]1,
 HELPTEXT="Logarithm (base 2) of alphabet size",PARNAME="Log2_of_M",
  PARSYM="@log2_of_alphabet_size@"
.PARAMETER,TYPE=INTEGER,VALUE=[10]5,
 HELPTEXT="Minimum errors to collect at each parameter iteration.",
  PARNAME="Requested_Minimum_Errors"
.PARAMETER,TYPE=INTEGER,VALUE=[100]@block@,
 HELPTEXT="Number of symbols processed per invocation.",
  PARNAME="Trial_Block_Size"
.PARAMETER,TYPE=INTEGER,VALUE=[1000]1000000000,
 HELPTEXT="Maximum symbols for any iteration before quitting.",
  PARNAME="Maximum_Symbols"
.PARAMETER,TYPE=INTEGER,VALUE=[5]5,HELPTEXT="Status Reporting Level",
 PARNAME="Reporting_Level"
.PARAMETER,TYPE=INTEGER,VALUE=[2]150,
 HELPTEXT="Number of steps in iterated parameter.",
  PARNAME="Num_Iterations"
.PARAMETER,TYPE=CHAR_STRING,VALUE=["Es/N0(dB)"]"*",
 HELPTEXT="Name of global symbolic parameter being iterated.",
  PARNAME="SNR_String"
.PARAMETER,TYPE=CHAR_STRING,VALUE=[".pbr"]"*",
```

```
 HELPTEXT="Filename of BER vs. SNR array.",PARNAME="BER_Filename"
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
 HELPTEXT="Initial symbols to discard before start of count.",
  PARNAME="Num_First_Discarded"
.INPUT,PORT_NUM=0,DATA_SIZE=4,DATA_TYPE=DIGITAL_DATA,
 SAMPLE_RATE=-2^(@log2_of_alphabet_size@)
.OUTPUT,PORT_NUM=0,DATA_SIZE=4,DATA_TYPE=DIGITAL_DATA,
 SAMPLE_RATE=-999.0
.ENDNODE
.MODULE,MODULE_NAME="berplt",INSTANCE_NAME="berplt",
CLASS_NAME="berplots",FILE_NAME="aco_plot\aco_plot",
 FUNC_NAME="aco_plot",NINPUTS=1,NOUTPUTS=0,NPARAMS=4,MAXACC=0,
  FBK_WORD=0,WORD_WIDTH=0,PRECISION=0,ROUND_CHAR=0,OVF_CHAR=0
.PLOT_INTERNALS,PLOT_ON_FLAG=FLAG_ON,
PLOT_COORDINATES={1558,6838,9555,22647},WIN_COORDINATES={-1,-1,-1,-1},
 REAL_TIME_FLAG=FLAG_ON,SCROLL_INCREMENT=0.000000,
  CURRENT_SCROLL_POINT=384.000000,RESCALE_ON_SCROLL_FLAG=FLAG_OFF,
   MIN_FILE_LENGTH_IN_UNITS=0,MAX_FILE_START_TIME=0.000000,
    MIN_FILE_STOP_TIME=0.000000
.PLOT_ANNOTATION,PLOT_TITLE="AUTO",
X_AXIS_NUMBER_EXTEND=FLUSH_ANNOTATION,X_AXIS_TEXT="AUTO",
 X_AXIS_NUMBER_FORMAT="f7.2",Y_AXIS_NUMBER_EXTEND=EXTEND_ANNOTATION,
  Y_AXIS_TEXT=" ",Y_AXIS_NUMBER_FORMAT="f7.2",
   R_AXIS_NUMBER_EXTEND=EXTEND_ANNOTATION,R_AXIS_TEXT="",
    R_AXIS_NUMBER_FORMAT="f7.2",NUMBER_OF_X_AXIS_TICS=8,
     NUMBER_OF_Y_AXIS_TICS=8,NUMBER_OF_R_AXIS_TICS=8,
      NUMBER_OF_X_GRID_CELLS=4,NUMBER_OF_Y_GRID_CELLS=-1,
       NUMBER_OF_R_GRID_CELLS=-1,LEGEND_FLAG=FLAG_OFF
.PLOT_FACTORS,PLOT_START_TIME=0.000000,PLOT_DURATION=128.000000,
SCROLL_DELTA=0.000000,SYMMETRIC_FLAG=FLAG_OFF,
 ABSOLUTE_X_AXIS_SCALING_FLAG=FLAG_OFF,
  ABSOLUTE_Y_AXIS_SCALING_FLAG=FLAG_OFF,
   LEFT_EDGE_SCALING_FACTOR=0.750000,RIGHT_EDGE_SCALING_FACTOR=0.750000,
    BOTTOM_EDGE_SCALING_FACTOR=0.750000,TOP_EDGE_SCALING_FACTOR=0.750000,
     LOWER_FFT_TRUNCATION_LIMIT=-100.000000,
      NORMALIZED_FFT_MAX_VALUE=0.000000
.PLOT_COLORS,X_AXIS_COLOR=GREEN,X_AXIS_NUMBER_COLOR=GREEN,
X_AXIS_TIC_COLOR=GREEN,X_AXIS_LABEL_COLOR=YELLOW,Y_AXIS_COLOR=GREEN,
 Y_AXIS_NUMBER_COLOR=YELLOW,Y_AXIS_TIC_COLOR=GREEN,
  Y_AXIS_LABEL_COLOR=YELLOW,R_AXIS_COLOR=BLACK,R_AXIS_NUMBER_COLOR=GREEN,
   R_AXIS_TIC_COLOR=GREEN,R_AXIS_LABEL_COLOR=YELLOW,
    PLOT_TITLE_COLOR=YELLOW,GRID_LINE_COLOR=GREY,BACKGROUND_COLOR=BLUE,
     BACKGROUND_OUTLINE_COLOR=YELLOW,GRID_BACKGROUND_COLOR=WHITE
.PLOT_CURVE,WAVEFILE_NAME="REAL_TIME",PLOT_TYPE=BIT_ERROR_RATE_PLOT,
CURVE_ANNOTATION="AUTO",PLOT_LINE_TYPE=NO_LINE,PLOT_LINE_THICKNESS=1,
 PLOT_LINE_COLOR=RED,PLOT_SYMBOL_CODE=STAR_SYMBOL,PLOT_SYMBOL_SIZE=300,
  PLOT_SYMBOL_COLOR=BLACK,PLOT_AXIS_USED=LEFT_AXIS
.PLOT_CURVE,WAVEFILE_NAME="BOUNDS_ETC:pskawgn.pbr",
PLOT_TYPE=BIT_ERROR_RATE_PLOT,CURVE_ANNOTATION="BPSKAWGN",
 PLOT_LINE_TYPE=SOLID_LINE,PLOT_LINE_THICKNESS=1,PLOT_LINE_COLOR=RED,
  PLOT_SYMBOL_CODE=NO_SYMBOL,PLOT_SYMBOL_SIZE=300,
   PLOT_SYMBOL_COLOR=YELLOW,PLOT_AXIS_USED=LEFT_AXIS
.ICON,ICON_SIZE={361,531},ICON_POSITION={245,595},ICON_ROTATION=0,
 SCREEN_STATE={0,0,1000,0},IPIN_SPACE=SPACED,IPIN_COOR={-16,0},
  OPIN_SPACE=SPACED,EXT_COLOR=RED,INT_COLOR=MAGENTA,
   BLOCK_NAME="BER%Plot",TEXT_COLOR=BLUE,PIN_SIZE=9,PIN_EXT_COLOR=MAGENTA,
    PIN_INT_COLOR=YELLOW,ICON_FILE="iconfile",ICON_INDEX=1
```

```
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
 HELPTEXT="Length in samples of data file",
  PARNAME="Scrollback_File_Length"
.PARAMETER,TYPE=INTEGER,VALUE=[100]100,
 HELPTEXT="Maximum number of real-time plots to generate.",
  PARNAME="Max_Plots"
.PARAMETER,TYPE=INTEGER,VALUE=[0]0,
 HELPTEXT="Set to non-zero for auto-scrollback and replot.",
  PARNAME="Replot_Count"
.PARAMETER,TYPE=INTEGER,VALUE=[1]1,
 HELPTEXT="Number of plots per call (-1 : run till exhausted).",
  PARNAME="Num_Plots_per_Call"
.INPUT,PORT_NUM=0,DATA_SIZE=4,DATA_TYPE=DIGITAL_DATA,
 SAMPLE_RATE=-999.0
.ENDNODE
.CONNECTION,FROM_NODE="dsrc",PORT_OUT=0,TO_NODE="xbpsk",PORT_IN=0,
 WIRE_SEGS={143,47,192,47},WIRE_COLOR=WHITE,WIRE_SIZE=1
.CONNECTION,FROM_NODE="xbpsk",PORT_OUT=0,TO_NODE="ssds",PORT_IN=0,
 WIRE_SEGS={258,47,320,47},WIRE_COLOR=WHITE,WIRE_SIZE=1
.CONNECTION,FROM_NODE="cwgn",PORT_OUT=0,TO_NODE="add",PORT_IN=0,
 WIRE_SEGS={617,86,563,86,563,117},WIRE_COLOR=WHITE,WIRE_SIZE=1
.CONNECTION,FROM_NODE="ssds",PORT_OUT=0,TO_NODE="add",PORT_IN=1,
 WIRE_SEGS={382,47,557,47,557,117},WIRE_COLOR=WHITE,WIRE_SIZE=1
.CONNECTION,FROM_NODE="unlu",PORT_OUT=0,TO_NODE="add",PORT_IN=2,
 WIRE_SEGS={488,91,549,91,549,116},WIRE_COLOR=WHITE,WIRE_SIZE=1
.CONNECTION,FROM_NODE="add",PORT_OUT=0,TO_NODE="ssds_1",PORT_IN=0,
 WIRE_SEGS={557,180,557,218,383,218},WIRE_COLOR=WHITE,WIRE_SIZE=1
.CONNECTION,FROM_NODE="ssds_1",PORT_OUT=0,TO_NODE="rbpsk",PORT_IN=0,
 WIRE_SEGS={320,218,294,218,294,215,270,215},WIRE_COLOR=WHITE,
  WIRE_SIZE=1
.CONNECTION,FROM_NODE="rbpsk",PORT_OUT=0,TO_NODE="sect",PORT_IN=0,
 WIRE_SEGS={197,224,177,224,177,215,158,215},WIRE_COLOR=WHITE,
  WIRE_SIZE=1
.CONNECTION,FROM_NODE="sect",PORT_OUT=0,TO_NODE="berplt",PORT_IN=0,
 WIRE_SEGS={95,215,34,215,34,595,65,595},WIRE_COLOR=WHITE,WIRE_SIZE=1
*
* END OF DS_AWGN.NET FILE
*
```

INITIAL DISTRIBUTION LIST

No. of copies

1. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6128

2. Dudley Knox Library 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, California 93943-5101

3. Chairman, Code EC 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943-5121

4. Professor Tri T. Ha, Code EC/Ha 2
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943-5121

5. Professor Vicente Garcia, Code EC/Ga 1
   Department of Electrical and Computer Engineering
   Naval Postgraduate School
   Monterey, California 93943-5121

6. CPT Yeu, Eng-Kiong 1
   Defense Technology Group
   23rd Storey, Tower A
   Defense Technology Towers
   Depot Road (0410)
   Republic of Singapore